



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**SISTEMA RADAR PARA RECONSTRUCCIÓN DE OBJETOS 3D**  
**MEDIANTE LA UTILIZACIÓN DEL DISPOSITIVO KINECT**

Trabajo de titulación presentado para optar el grado académico de:  
**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES**

**AUTOR/A: BARBERÁN MOREIRA JENEFFER JOSELIN**  
**DOMÍNGUEZ OROZCO RODRIGO FERNANDO**

**TUTOR:** Ing. Hugo Moreno Avilés.

Riobamba-Ecuador

2016

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA EN INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES**

El Tribunal del trabajo de titulación certifica que: El trabajo de titulación: **SISTEMA RADAR PARA RECONSTRUCCIÓN DE OBJETOS 3D MEDIANTE LA UTILIZACIÓN DEL DISPOSITIVO KINECT**, de responsabilidad de la señorita Jeneffer Joselin Barberán Moreira y el señor Rodrigo Fernando Domínguez Orozco , ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación quedando autorizada su presentación.

**DR. MIGUEL TASAMBAY**  
**DECANO DE LA FACULTAD DE**  
**INFORMÁTICA Y ELECTRÓNICA**

\_\_\_\_\_

**ING. FRANKILN MORENO**  
**DIRECTOR DE LA ESCUELA DE**  
**INGENIERIA ELECTRÓNICA,**  
**TELECOMUNICACIONES Y REDES**

\_\_\_\_\_

**ING. HUGO MORENO**  
**DIRECTOR DE TESIS**

\_\_\_\_\_

**ING. IRENE TUSTÓN**  
**MIEMBRO DEL TRIBUNAL**

\_\_\_\_\_

Nosotros, Jeneffer Joselin Barberán Moreira y Rodrigo Fernando Domínguez Orozco, declaráramos ser los autores del presente trabajo de titulación “: **SISTEMA RADAR PARA RECONSTRUCCIÓN DE OBJETOS 3D MEDIANTE LA UTILIZACIÓN DEL DISPOSITIVO KINECT**”, que fue elaborado en su totalidad por nosotros, bajo la dirección del Ingeniero Hugo Moreno Avilés, haciéndonos totalmente responsables por las ideas, criterios, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio de la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

---

JENEFFER JOSELIN BARBERÁN MOREIRA

---

RODRIGO FERNANDO DOMÍNGUEZ OROZCO

## **DEDICATORIA**

A Dios por haberme bendecido cada momento de mi vida estudiantil, a mis padres quienes son pilares fundamentales de mi vida y han sido la guía y el camino para llegar a culminar mi carrera, a mi esposa Dany y mi Hija Danna quienes son la razón de vivir cada día con mucho amor, felicidad y responsabilidad, a mis hermanos Juan, Carina, Mónica, María, Carlos y John por compartir conmigo muchas alegrías y fracasos. Les dedico todo mi esfuerzo en reconocimiento a todo el sacrificio puesto en mí para cumplir esta meta, ustedes mi querida familia se merecen esto y mucho más, con amor

Rodrigo

Este trabajo se lo dedico a Dios quien supo guiarme por el buen camino, darme fuerza para seguir adelante y no desmayar en los problemas que se presentan, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. A mi familia quienes por ello soy lo que soy. Para mis padres, mi esposo Franklin y mi hijo Jherath por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para seguir mis objetivos.

Jeneffer

## **AGRADECIMIENTO**

A mi familia fuente de apoyo constante e incondicional en toda mi vida y más aún en mis duros años de carrera profesional, en especial quiero expresar mi más grande agradecimiento a mis padres que sin su ayuda hubiera sido imposible culminar mi profesión.

De igual manera a mi esposo Franklin y mi hijo Jherath, quienes son mi orgullo y gran motivación logrando impulsarme cada día a superarme en mi carrera profesional, quienes han estado en todo momento junto a mí.

Jeneffer

El más sincero agradecimiento a mis padres que a pesar de cualquier inconveniente, se han fortalecido para permitirnos estudiar a mis hermanos y a mí, además de brindarme su amor incondicional, también agradezco a Daniela por acompañarme en todo momento desde nuestra unión, a mis hermanos y al Ingeniero Hugo Moreno quien nos ha guiado de manera muy amable en nuestro trabajo de titulación.

Rodrigo.

## ÍNDICE GENERAL

PORTADA.....	i
FIRMA DE RESPONSABILIDADES.....	ii
RESPONSABILIDAD DE AUTORES.....	ii
DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
INDICE DE FIGURAS.....	ix
INDICE DE TABLAS.....	xii
INDICE DE ECUACIONES.....	xiii
INDICE DE ANEXOS.....	xiv
INDICE DE ABREVIATURAS.....	xv
RESUMEN.....	xvii
SUMMARY .....	xviii
INTRODUCCION.....	1

## CAPITULO I

### 1 MARCO TEORICO

1.1 Sistema Radar .....	6
1.1.1 Definición.....	6
1.1.2 Funcionamiento del Sistema Radar.....	6
1.1.3 Tipos de Radares .....	8
1.1.3.1 Radar de Impulso .....	8
1.1.3.2 Radar de Onda Continua.....	9
1.1.4 Frecuencias de Operación.....	9
1.1.5 Radar Laser (LADAR).....	10
1.1.6 Reconstrucción 3D.....	13
1.1.7 Adquisición de Datos .....	13
1.1.7.1 Tiempo de Vuelo 3D o Telemetría.....	14
1.1.7.2 Triangulación Actica.....	15
1.1.7.3 Luz Estructurada .....	16
1.1.8 Análisis del Sistema Radar .....	17
1.1.8.1 Resolución del Sistema .....	17

1.1.8.2	<i>Distancia Máxima de Análisis de Señal</i> .....	17
1.1.9	<i>Detección del Objeto</i> .....	18
1.1.10	<i>Distancia del Objeto Detectado</i> .....	19
1.2	<b>Kinect</b> .....	20
1.2.1	<i>Cámara RGB</i> .....	21
1.2.2	<i>Sensor de Profundidad</i> .....	22
1.2.3	<i>Matriz de Micrófono</i> .....	23
1.2.4	<i>Chip PrimeSense</i> .....	24
1.2.5	<i>Motor de Inclinación</i> .....	25
1.2.6	<i>Especificaciones Técnicas</i> .....	25
1.2.7	<i>Controladores y Plataformas</i> .....	25
1.2.7.1	<i>Controladores OpenKinect</i> .....	26
1.2.7.2	<i>Controlador OpenNI/NITE</i> .....	26
1.2.7.3	<i>Controlador Microsoft Kinect SDK</i> .....	28

## CAPITULO II

### 2 SISTEMA RADAR PARA RECONSTRUCCION DE OBJETOS 3D MEDIANTE EL DISPOSITIVO KINECT

2.1	<b>Sistema Radar</b> .....	30
2.1.1	<i>Funcionamiento Sistema Radar</i> .....	30
2.1.2	<i>Librerías</i> .....	31
2.2	<b>Código Matlab</b> .....	31
2.2.1	<i>Enlace Matlab con Sensor Kinect</i> .....	32
2.2.2	<i>Selección de Información</i> .....	33
2.2.3	<i>Almacenamiento de Información</i> .....	34
2.2.4	<i>Reconstrucción Del Objeto En 3D</i> .....	42
2.2.4.1	<i>Creación de las Matrices 0's</i> .....	42
2.2.4.2	<i>Matriz Binaria</i> .....	43
2.3	<b>Diseño de la Aplicación GUI</b> .....	51

## CAPITULO III

### 3 MEDICIONES Y RESULTADOS

3.1	<b>Objetos de Reconstrucción</b> .....	54
3.2	<b>Parámetros</b> .....	54
3.2.1	<i>Matrices de Datos Primer Objeto</i> .....	55

3.2.1.1	<i>Matrices de Datos Primer Objeto Prueba 1</i>	56
3.2.1.2	<i>Matrices de Datos Primer Objeto Prueba 2</i>	59
3.2.2	<b><i>Matrices de Datos Segundo Objeto</i></b>	62
3.2.2.1	<i>Matrices de Datos Primer Objeto Prueba 1</i>	63
3.2.2.2	<i>Matrices de Datos Segundo Objeto Prueba 2</i>	67
3.3.	<b>Reconstrucción de Objetos en 3D</b>	72
3.3.1	<b><i>Reconstrucción 3D Primer Objeto</i></b>	72
3.3.2	<b><i>Reconstrucción 3D Segundo Objeto</i></b>	73
3.4	<b>Análisis Comparativo</b>	75
3.4.1	<b><i>Análisis Objeto 1</i></b>	75
3.4.2	<b><i>Análisis Objeto 2</i></b>	77
3.5	<b>Concatenación de Datos</b>	78
3.5.1	<b><i>Reconstrucción Objeto 1 con Concatenación</i></b>	79
3.5.2	<b><i>Reconstrucción Objeto 2 con Concatenación</i></b>	79
3.6	<b>Resolución y Precisión</b>	84
	<b>CONCLUSIONES</b>	86
	<b>RECOMENDACIONES</b>	87
	<b>BIBLIOGRAFIA</b>	88
	<b>ANEXOS</b>	



## ÍNDICE DE FIGURAS

FIGURA 1-1:	Funcionamiento Sistema Radar.....	7
FIGURA 2-1:	Forma Ideal de Señal Transmitida por un Radar de Impulso.....	8
FIGURA 3-1:	Frecuencia de Operación del Radar.....	10
FIGURA 4-1:	Radares con su Banda de Operación.....	11
FIGURA 5-1:	Configuración Bi-estática en el Sistema LADAR/RADAR.....	12
FIGURA 6-1:	Configuración Mono-estática en el Sistema LADAR/RADAR.....	12
FIGURA 7-1:	Principio de Tiempo de Vuelo .....	14
FIGURA 8-1:	El Error Tangencial .....	14
FIGURA 9-1:	Principio de Triangulación .....	15
FIGURA 10-1:	Principio de Funcionamiento del Escáner de Luz Estructurada .....	16
FIGURA 11-1:	Señal Transmitida – Reflejada .....	17
FIGURA 12-1:	Transformada Rápida de Fourier.....	18
FIGURA 13-1:	Transformada Rápida de Fourier.....	20
FIGURA 14-1:	Componentes del Kinect .....	21
FIGURA 15-1:	Ángulos de Limitación .....	22
FIGURA 16-1:	Método de Luz Estructurada .....	22
FIGURA 17-1:	Patrón de Puntos Proyectado.....	23
FIGURA 18-1:	Matriz de Micrófono .....	24
FIGURA 19-1:	Proceso de Adquisición de Datos de la Cámara RGB.....	24
FIGURA 20-1:	Angulo del Motor de Inclination .....	25
FIGURA 21-1:	Captura de Profundidad mediante Controlador OpenKinect.....	27
FIGURA 22-1:	Esqueletización y Seguimiento de una Persona mediante OpenNI.....	27
FIGURA 23-1:	Reconocimiento de Personas mediante Microsoft Kinect SDK.....	28
FIGURA 1-2:	Sistema Radar utilizando el Dispositivo Kinect.....	30
FIGURA 2-2:	Diagrama de Flujo Etapas de Código Matlab .....	32
FIGURA 3-2:	Imagen RGB e Imagen de Profundidad tomado con el Sensor Kinect .....	33
FIGURA 4-2:	Obtención de Información.....	34
FIGURA 5-2:	Parámetros para la Reconstrucción 3D .....	34
FIGURA 6-2:	Flujograma para Almacenar Matriz de Información.....	37
FIGURA 7-2:	Flujograma para Almacenar Matrices de Información.....	38

FIGURA 8-2:	Asignación de 1's en Matrices 0's. ....	46
FIGURA 9-2:	Obtención de Datos en forma Horaria o Anti-Horaria .....	47
FIGURA 10-2:	Rotación de Matrices Binarias. ....	47
FIGURA 11-2:	Matriz Binaria. ....	48
FIGURA 12-2:	Reconstrucción del Objeto en 3D.....	49
FIGURA 13-2:	Flujograma para Reconstrucción del Objeto en 3D. ....	50
FIGURA 14-2:	Componentes de la Interfaz Gráfica de Usuario .....	51
FIGURA 15-2:	Callback.....	52
FIGURA 16-2:	Ventana Script para Codificación de Componentes.....	52
FIGURA 17-2:	GUI Sistema Radar para Reconstrucción de Objetos en 3D .....	53
FIGURA 1-3:	Objeto en forma Rectangular .....	54
FIGURA 2-3:	Objeto 1.....	55
FIGURA 3-3:	Matrices de 0's con Asignación de 1's Primer Objeto Prueba 1 .....	58
FIGURA 4-3:	Asignación de 1's en la Matriz de 0's Primer Objeto Prueba 2 .....	61
FIGURA 5-3:	Objeto 2.....	62
FIGURA 6-3:	Objeto 2 Cara Posterior .....	62
FIGURA 7-3:	Matrices 0's Reasignado 1's Primera Fila de Matrices Almacenadas Prueba 1.....	65
FIGURA 8-3:	Matrices 0's Reasignado 1's Segunda Fila de Matrices Almacenadas Prueba 1.....	66
FIGURA 9-3:	Matrices 0's Reasignado 1's Primera Fila de Matrices Almacenadas Prueba 2.....	70
FIGURA 10-3:	Matrices 0's Reasignado 1's Segunda Fila de Matrices Almacenadas Prueba 2.....	71
FIGURA 11-3:	Reconstrucción Objeto 1 Prueba 1.....	72
FIGURA 12-3:	Reconstrucción Objeto 1 Prueba 2.....	73
FIGURA 13-3:	Reconstrucción Objeto 2 Prueba 1.....	73
FIGURA 14-3:	Reconstrucción Objeto 2 Prueba 1 Vista 2.....	74
FIGURA 15-3:	Reconstrucción Objeto 2 Prueba 2.....	74
FIGURA 16-3:	Reconstrucción Objeto 2 Prueba 2 Vista 2.....	75
FIGURA 17-3:	Objeto 1 Real y Reconstrucción 3D Prueba 1 .....	76
FIGURA 18-3:	Objeto 1 Real y Reconstrucción 3D Prueba 2.....	76

FIGURA 19-3:	Objeto 2 Real y Reconstrucción 3D Prueba 1 .....	77
FIGURA 20-3:	Objeto 2 Real y Reconstrucción 3D Prueba 2 .....	78
FIGURA 21-3:	Reconstrucción 3D Objeto 1 con Datos Concatenados Tabla 3-3 .....	80
FIGURA 22-3:	Reconstrucción 3D Objeto 1 con Datos Concatenados Tabla 5-3 .....	80
FIGURA 23-3:	Reconstrucción 3D Objeto 2 con Datos Concatenados Tabla 7-3 .....	81
FIGURA 24-3:	Reconstrucción 3D Objeto 2 con Datos Concatenados Tabla 9-3 .....	82
FIGURA 34-3:	Flujograma para Concatenación de Datos.....	83

## ÍNDICE DE TABLAS

TABLA 1-1:	Características del Kinect.....	26
TABLA 2-1:	Ventajas y Desventajas entre el Controlador OPENNI y Microsoft .....	29
TABLA 1-3:	Parámetros .....	55
TABLA 2-3:	Matriz de Datos medidos del Primer Objeto Prueba 1 .....	56
TABLA 3-3:	Cantidad de 1´s Prueba 1 .....	57
TABLA 4-3:	Matriz de Datos medidos del Primer Objeto Prueba 2 .....	59
TABLA 5-3:	Cantidad de 1´s Prueba 2.....	60
TABLA 6-3:	Matriz de Datos medidos del Segundo Prueba 1.....	63
TABLA 7-3:	Cantidad de 1´s Prueba 1 Objeto 2 .....	64
TABLA 8-3:	Matriz de Datos medidos del Segundo Objeto Prueba 2 .....	67
TABLA 9-3:	Cantidad de 1´s Prueba 2 Objeto 2.....	68
TABLA 10-3:	Error de Profundidad .....	84

## ÍNDICE DE ECUACIONES

Ecuación 1-1:	Distancia al Blanco .....	8
Ecuación 2-1:	Ancho de Banda.....	17
Ecuación 3-1:	Resolución del Sistema.....	17
Ecuación 4-1:	Distancia Máxima Teórica.....	17
Ecuación 5-1:	Distancia Máxima Practica .....	17
Ecuación 6-1:	Señal Transmitida .....	18
Ecuación 7-1:	Señal Recibida .....	18
Ecuación 8-1:	Señal Chirp Transmitida .....	19
Ecuación 9-1:	Frecuencia Variante en el Tiempo .....	19
Ecuación 10-1:	Rapidez en el Cambio de Frecuencia.....	19
Ecuación 11-1:	Retardo en la señal Chirp transmitida y recibida .....	19
Ecuación 12-1:	Posición del Objeto Detectado.....	19
Ecuación 1-2:	Número de Datos en Vector.....	36
Ecuación 2-2:	Número de Filas.....	36
Ecuación 3-2:	Número de Columnas .....	36
Ecuación 4-2:	Posición del Objeto .....	43
Ecuación 5-2:	Posición del Objeto Primera Cara.....	44
Ecuación 6-2:	Posición del Objeto Segunda Cara.....	44
Ecuación 7-2:	Posición del Objeto Tercera Cara .....	45
Ecuación 8-2:	Posición del Objeto Cuarta Cara.....	45
Ecuación 9-2:	Matriz Binaria Resultante .....	48
Ecuación 10-2:	Matriz Binaria General .....	48
Ecuación 1-3:	Error Verdadero .....	84
Ecuación 2-3:	Error Relativo Porcentual .....	84
Ecuación 3-3:	Media del Error Relativo Porcentual .....	84
Ecuación 4-3:	Formula de la Precisión .....	85
Ecuación 5-3:	Reemplazo Formula de la Precisión .....	85
Ecuación 6-3:	Precisión.....	85

## ÍNDICE DE ANEXOS

ANEXO A:	Instalación del Dispositivo Kinect .....
ANEXO B:	Código Sistema Radar .....
ANEXO C:	Flujograma Sistema Radar Para Reconstruir Objeto en 3D .....

## INDICE DE ABREVIATURAS

<b>API</b>	Application Programming Interface (Interfaz de Programación de Aplicaciones)
<b>CIF</b>	Common Intermediate Format (Formato Intermedio Común)
<b>CMOS</b>	Complementary metal-oxide-semiconductor (Semiconductor complementario de óxido metálico)
<b>FFT</b>	Fast Fourier Transform (Transformada Rápida de Fourier)
<b>FPS</b>	Frames per Second (Cuadros por segundo)
<b>GUI</b>	Interfaz Gráfica de Usuario
<b>HF</b>	High Frequency
<b>IEEE</b>	Instituto de Ingenieros Eléctricos y Electrónicos
<b>LADAR</b>	Láser Detección And Ranging (Detección y medición de distancias por Laser)
<b>LIDAR</b>	Light Detection and Ranging (Detección y medición de distancias por Luz)
<b>NIST</b>	Instituto Nacional de Estándares y Tecnología
<b>RADAR</b>	Radio Detection And Ranging (Detección y medición de distancias por radio)
<b>RGB</b>	Red Green Blue (Rojo Verde Azul)
<b>RF</b>	Radio Frecuencia
<b>UHF</b>	High Frequency (Frecuencia Ultra Alta)
<b>USB</b>	Universal Serial Bus
<b>VHF</b>	Very High Frequency (Muy alta frecuencia)
<b>VGA</b>	Video Graphics Array ó Video Graphics Adapter (Arreglo gráfico de video ó adaptador gráfico de video)
<b>3D</b>	Three Dimensional (Tres dimensiones)
<b>CCD</b>	<b>Charge-Coupled Device</b> (Dispositivo de carga acoplada)
<b>BANDA X</b>	Rango de frecuencias entre 8,2 y 12,4 GHz
<b>BANDA K</b>	Banda de frecuencias entre 20 y 40 GHz
<b>BANDA C</b>	Frecuencias entre 3,7 y 4,2 GHz y desde 5,9 hasta 6,4 GHz.
<b>BANDA S</b>	Rango frecuencial desde 1,5 GHz a 5,2 GHz

<b>BANDA Ka</b>	Opera entre los 26,5 Ghz y 40 GHz
<b>BANDA Ku</b>	(Kurz-unten band) rango de las microondas desde 12 GHz a los 18 GHz
<b>FM</b>	Modulated Frequency (Frecuencia modulada)
<b>ADC</b>	Convertidor Analógico Digital
<b>IR</b>	Infrared (Radiación Infrarroja)
<b>SDK</b>	Software Development Kit (kit de desarrollo de software )



## RESUMEN

Se diseñó y evaluó un sistema radar para la reconstrucción de objetos en 3D mediante el dispositivo Kinect para el grupo de investigación de electromagnetismo y microondas que posee la Facultad de Informática y Electrónica en la Escuela Superior Politécnica de Chimborazo de la ciudad de Riobamba. La metodología utilizada en este trabajo de investigación es cuantitativa exploratoria puesto que se desarrolló en cinco bloques principales, representando los procesos inherentes al manejo y uso de la información proporcionada por el dispositivo Kinect. Estos bloques son: Capturar la Imagen, Estimación de Profundidad, Almacenar Información, Representar profundidad en matrices cuadradas binaria y Reconstrucción 3D. Las herramientas utilizadas para el funcionamiento del Sistema Radar constan del dispositivo Kinect, computador, controlador Open NI y software Matlab. Los resultados obtenidos experimentalmente luego de las mediciones realizadas con objetos lineales, determinó que la medida de profundidad estimada por el Sensor Kinect tiene una Resolución de 3mm y la Digitalización del Objeto tiene una precisión de 99.88%. A través de las pruebas realizadas para reconstruir objetos en un área de  $(2 \times 2) \text{ m}^2$ , se concluyó que la distancia máxima del Kinect al objeto debe ser de 1.2m y la distancia mínima 0.5m siempre y cuando la cámara IR se encuentre paralela al eje del suelo. El algoritmo implementado en Matlab permite a los objetos digitalizarlos a partir de matrices binarias donde cada elemento con valor 1 representa una fracción del objeto. Se recomienda que para futuras investigaciones en reconstrucción tridimensional, aumentar el número de vistas para la toma de mediciones con la finalidad de obtener mayor cantidad de datos que permitan aumentar la precisión en la reconstrucción de objetos.

**Palabras Claves:** <OBJETOS EN TRES DIMENSION [3D]>, <SOFTWARE [Matlab]>, <CONTROLADOR [Open NI]>, <DISPOSITIVO [Kinect]>, <RECONSTRUCCIÓN DE OBJETOS>, < MATRICES BINARIAS>, < CAMARA DE PROFUNDIDAD [IR]>.

## SUMMARY

It was designed and evaluate a radar system for the reconstruction of 3D objects using the Kinect device for the research group electromagnetism and microwave that has the faculty of computer science and electronics in Escuela Superior Politécnica de Chimborazo in Riobamba city. The methodology used in this research is exploratory quantitative since it was developed in five principals blocks, representing the processes inherent in the management and use of the information provided by the Kinect device. These blocks are: Capture the Image, Depth Estimation, Store Information, Representing Depth in Binary Square matrices, and 3D Reconstruction. The tools used for the operations of Radar System consists of Kinect device, computer, Open NI controller, and Matlab Software. Experimental results after measurements made with linear objects, determine the estimated depth measurement by the sensor Kinect has a 3 mm Resolution and Object Scanning has an accuracy of 99.88%. Through tests conducted to reconstruct objects in an area (2×2) m<sup>2</sup>, it was concluded that the maximum distance of the Kinect to Object should be the 1.2m and the minimum distance 0.5m as long as the IR camera is parallel to the axis of the floor. The algorithm implemented in Matlab allows digitizing objects form binary matrix where each element value 1 has a fraction of the object. It is recommended that future research on three-dimensional reconstruction, increase the number of views for making measurements in order to obtain as much data to increase the accuracy in objects reconstruction.

**Keywords:** <OBJETS IN THREE DIMENSIONS [3D]>, <SOFTWARE [Matlab]>, <CONTROLLER [Open NI]>, <DEVICE [Kinect]>, <RECONSTRUCTIONS OF OBJECTS>, < MATRIX BINARY>, < CAMERA DEPTH [IR]>.

## INTRODUCCION

En la actualidad la representación de objetos en forma tridimensional ocupa un amplio campo en el desarrollo tecnológico, permitiendo al usuario obtener una representación digital de un objeto, basándose en la captura de imágenes con una cámara web y un láser, que posteriormente son almacenadas para pasar a la etapa de procesamiento. Esta reconstrucción tridimensional resulta errónea al momento de representar digitalmente una figura compleja, esto sucede debido a la utilización de una sola cámara para capturar imágenes de secciones dificultosas, impidiendo la reconstrucción total del objeto<sup>1</sup>.

La presente investigación tiene como objetivo: Diseñar y evaluar un sistema radar para reconstrucción de objetos 3D mediante la utilización del dispositivo Kinect para aquello es necesario analizar el funcionamiento del sistema radar mediante el sensor Kinect, desarrollar un algoritmo mediante el procesamiento de señales para la reconstrucción de objetos 3D, evaluar el Algoritmo mediante el Software Matlab, analizar resultados del proceso de evaluación y optimizar el algoritmo mediante pruebas de campo.

Las metodologías que se implementarán en esta investigación son: investigación bibliográfica, desarrollo de software, construcción total y documentación del proyecto. Este proyecto será un aporte para las múltiples ramas de la ingeniería, que estén interesados por el tema de escáneres tridimensionales para la reconstrucción de cualquier tipo de objetos que posteriormente servirá como herramienta para la elaboración de nuevos elementos debido a que en la actualidad existen impresoras 3D, de tal forma que este sistema sea una herramienta para la colectividad.

---

<sup>1</sup> **YÁNEZ, Jorge.** Diseño y Construcción de un Escáner Bifocal para la Reconstrucción Visual de Objetos utilizando Procesamiento de Imágenes mediante Filtrado en Labview y Matlab. TESIS. Universidad Politécnica Salesiana. Facultad de Ingeniería Electrónica. Quito-Ecuador. 2014.

## ANTECEDENTES

El radar apareció a inicios del siglo XX, de manera que los científicos anteriormente tuvieron que comprender los principios de las ondas de radio. Fue en 1887, cuando el físico Heinrich Hertz empezó analizar las ondas de radio, lo que creó la iniciativa de muchos científicos en la búsqueda de una aplicación por medio de las ondas de radio. Sin embargo fue en 1935 que Sir Robert Watson Watt logró demostrar la detección de un avión por un aparato de radio llamado experimento “Daventry”. Dando el inicio a la primera red de radares lo cual denominaron “Chain Home” <sup>2</sup>.

El Radar ha ido evolucionando gracias al progreso tecnológico, dichos avances han permitido construir sistemas altamente complejos desde radares tridimensionales capaces de situar y seguir centenares de blancos en distancia, azimuth y elevación, o radares transhorizonte que trabajan en HF teniendo alcance del orden de 2000 km. Hasta sistema radares para el sondeo geológico subterráneo o radares láser para la medida de aerosoles y contaminantes en la atmósfera <sup>3</sup>.

Hoy en día el sistema Radar es una rama de la electrónica de mayor impacto, ha permitido crear numerosas aplicaciones, sobre todo en fines militares dando el impulso a otros campos enfocados a dar soluciones a problemáticas, como en la agricultura, geología, cartografía incluso en la meteorología permitiendo predecir el comportamiento de eventos como fuertes tormentas, tornados, granizadas, lluvias, etc <sup>2</sup>.

En nuestro país existen proyectos como el nuevo sistema de radares de control para el tráfico aéreo que la Aviación Civil implementará. Se trata de un sistema integrado que proveerá el 95% de control aéreo desde todos los puntos del país, con un dispositivo en los aviones, permitiendo la visualización del espacio aéreo durante todo el desarrollo de la nave desde su despegue hasta su aterrizaje <sup>4</sup>.

---

<sup>2</sup> **SZYMANCZYK, OSCAR.** *Historia de las Telecomunicaciones Sistema*. 1ed. Argentina - Buenos Aires. Dunken. 2013. p. 171

<sup>3</sup> **PEREZ, GUTIERREZ. LUIS, MUÑOZ.** *Teledetección: nociones y aplicaciones*. 1a ed., España, - Salamanca. 2006. p. 223

<sup>4</sup> *Ecuador tiene el sistema de radares más moderno de Latinoamérica -Lenguas Ancestrales Amazónicas-* [en línea]. Ecuador. [Consulta: 16 Junio 2015]. Disponible en: <http://www.desarrolloamazonico.gob.ec/ecuador-tiene-el-sistema-de-radares-mas-moderno-de-latinoamerica/>

## **FORMULACIÓN DEL PROBLEMA**

La reconstrucción de objetos de forma tridimensional se lo puede realizar por medio de Scanner 3D, de manera que se puede mencionar escáners como el HandyScan, Láser Escáner de Leica HDS30003 los cuales permiten el relevamiento de volúmenes en tres dimensiones. Muchos requieren del servicio de escaneando 3D, pero pocos son los lugares disponibles con este evolucionado capital tecnológico. Además de complementarse con la maquinaria existente en fábrica, este tipo de tecnología impulsa la creación de diversas unidades de negocio.

El proceso en el escaneo de objetos 3D resulta costoso en el país razón por la cual en el presente proyecto se proveerá un sistema radar para la reconstrucción de objetos 3D a bajo costo mediante la utilización de elementos disponibles en el mercado como el Kinect

## **JUSTIFICACIÓN TEÓRICA**

El presente proyecto de investigación pretende aportar a la reconstrucción tridimensional de objetos y escenas debido a que es un tema ampliamente investigado por la necesidad de crear mecanismos capaces de reconstruir escenas 3D basada en la abstracción de imágenes secuenciales del mundo real con el fin de obtener una réplica del objeto lo más parecida al real. Existen métodos tales como RGB, Estereográfico, Time-of-light y de luz estructurada que buscan determinar representaciones volumétrica de objetos cuyo objetivo principal es obtener un algoritmo que sea capaz de realizar la conexión del conjunto de puntos representativos del objeto en forma de elementos de superficie, de manera que al entrelazar la información extraída de la captura de imágenes del objeto se le da forma al elemento.

Actualmente los algoritmos se debaten entre el coste computacional y la calidad del mallado obtenido en la reconstrucción de objetos en 3D. Existen estudios acerca de la reconstrucción de objetos bidimensionales a través de sensores ópticos como “Sistema radar para el análisis y reconstrucción de perfiles topográficos” realizado por alumnos de la Escuela Superior Politécnica de Chimborazo, en el cual se determinó la sensibilidad del Kinect que es de  $\pm 1.5$  mm, siendo como punto de partida dicho documento para realizar nuestra implementación orientado al desarrollo de un sistema radar para reconstruir objetos en 3D.

Se investiga fundamentos que determinen un algoritmo que permita representar computacionalmente objetos 3D utilizando el dispositivo de imágenes RGB-D debido a que es

un dispositivo que tiene una cámara de profundidad que usa el método de luz estructurada el cual permite obtener datos obtenidos directamente de las imágenes, reconstruyendo la forma y características de un objeto mediante información adquirida de la proyección de un patrón de luz. Se Investigará la comunicación hombre-máquina para la utilización del hardware y software apropiado para procesar la información de la captura secuencial de imágenes hechas por el Kinect de un objeto en particular con el fin de analizar y determinar el algoritmo requerido para reconstruir el objeto.

Para poder graficar el objeto en forma tridimensional que permita obtener información de una distancia requerida para la reconstrucción del objeto, aprovecharemos el sensor de profundidad láser que posee el dispositivo RGB-D, donde analizaremos los requerimientos necesarios para que esta información sea vinculado a un ordenador computacional, de manera que adaptaremos el sensor de profundidad a la computadora utilizando así herramientas de software para el estudio de la información.

La investigación determinará un algoritmo para la reconstrucción de objetos 3D generando un software que procese la información mediante el algoritmo estructurado, cubriendo así el estudio para la obtención tridimensional de objetos a diferentes escalas, de la misma manera implementaremos el sistema radar cuyos parámetros serán analizados desde el punto vista matemático y físico con la finalidad de optimizar el algoritmo que permite reconstruir objetos planares.

## **JUSTIFICACIÓN APLICATIVA**

Se implementara un sistema radar que consta de la fusión del dispositivo Kinect y un sistema computacional por medio de la interfaz OpenNI, permitiendo el procesamiento de señales para la reconstrucción de objetos 3D. El periférico Kinect consta de un sensor infrarrojo que proveerá una nube de puntos los cuales serán procesados por un filtro para estimar la superficie del objeto. El sistema computacional consta del software Matlab y los archivos Mex, que proporcionarán la información necesaria para desarrollar el algoritmo de reconstrucción tridimensional.

El sistema radar logrará impulsar el desarrollo de aplicaciones en múltiples ramas de la ingeniería como por ejemplo en el campo de la medicina permitiendo reconstruir órganos humanos lo que ayudará en el estudio del cuerpo humano, además se podrá utilizar este sistema

radar para reconstruir cualquier tipo de objetos que posteriormente servirá como herramienta para la elaboración de nuevos elementos debido a que en la actualidad existen impresoras 3D.

Un ejemplo de esto puede ser la reconstrucción de adornos de casa, que son elementos importados del exterior por tal motivo su valor para adquirir es elevado sin embargo el presente proyecto permitirá realizar la reconstrucción de objetos 3D mediante dispositivos que se encuentran en el mercado de nuestro país como es el Kinect, reduciendo los valores de adquisición y de esta manera mejorar la economía.

El beneficio que brindará el proyecto será el de determinar un sistema radar capaz de reconstruir objetos tridimensionalmente de tal forma que este sistema sea una herramienta para la colectividad.

## **OBJETIVOS**

### **OBJETIVOS GENERALES**

Diseñar y evaluar un sistema radar para reconstrucción de objetos 3D mediante la utilización del dispositivo Kinect

### **OBJETIVOS ESPECÍFICOS**

- Analizar el funcionamiento del sistema radar mediante el sensor Kinect.
- Desarrollar un algoritmo mediante el procesamiento de señales para la reconstrucción de objetos 3D.
- Evaluar el Algoritmo mediante el Software Matlab.
- Analizar resultados del proceso de evaluación.
- Optimizar el algoritmo mediante pruebas de campo.

## **CAPITULO I**

### **1. MARCO TEORICO**

#### **1.1. SISTEMA RADAR**

##### **1.1.1. Definición**

El Sistema Radar adopta su nombre del acrónimo Radio Detection And Ranging (“Detección y localización por radio”) <sup>5</sup>. De manera que si lo definimos por la función que desempeñan y la manera de cómo lo hacen, diríamos que son sistemas cuyo objetivo es descubrir la presencia de blancos a una cierta distancia gracias a las radiaciones electromagnéticas que permite detectar la localización o la velocidad de un objeto a través de ecos producidos por las radiaciones en respuesta a la emisión de señales electromagnéticas.

El Sistema Radar tiene como proceso transmitir, recibir, detectar y procesar una onda electromagnética que se refleja desde un objetivo y permite determinar la posición, la velocidad y las características de objetos que se encuentran lejos del alcance visual.

Sus aplicaciones se han diversificado enormemente, lo usan los meteorólogos para detectar tormentas, huracanes y tornados, los controladores aéreos para ordenar el tráfico de los aeropuertos, la NASA para crear mapas topográficos de los planetas y la policía para determinar la velocidad a la que circulan los vehículos <sup>6</sup>.

##### **1.1.2. Funcionamiento del Sistema Radar.**

El principio de funcionamiento de los radares se basa en dos sencillos fenómenos físicos: El eco y el efecto Doppler.

---

<sup>5</sup> **ÁLVAREZ, R & SHAGÑAY, S.** *Sistema Radar Para Análisis Y Reconstrucción De Perfiles Topográficos A Media Escala*. 2015: pp.18

<sup>6</sup> **FLORES, Mario.** *Sistema de Radar* [blog]. [Consulta: 3 diciembre 2015] Disponible en: <https://webtelecomm.wordpress.com/2015/01/24/sistemas-de-radar/>

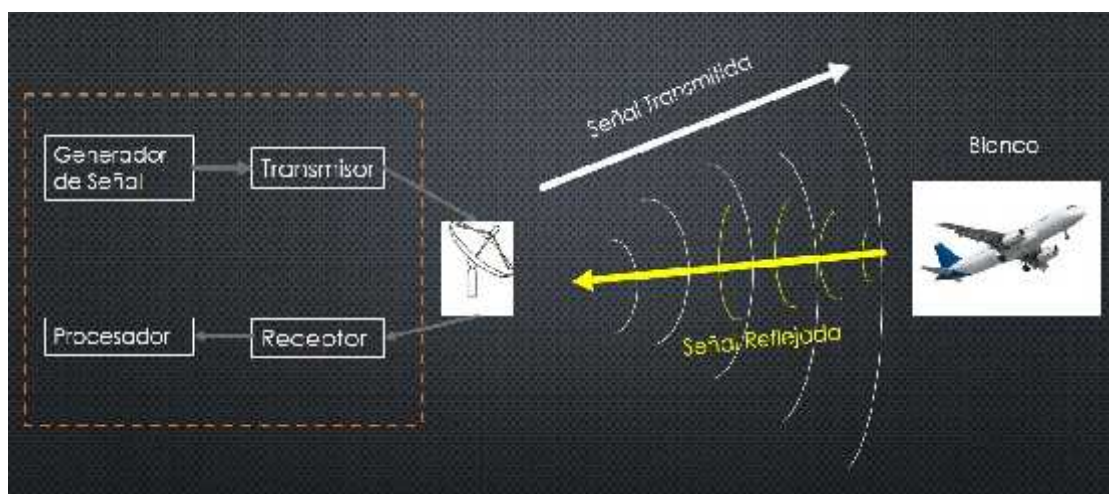


- **Eco:** Al igual que un sonido, cuando una onda electromagnética que se propaga por el aire choca contra un obstáculo, parte de su energía es absorbida y parte reflejada hacia el emisor.
- **Efecto Doppler:** El efecto doppler consiste en la variación de frecuencia de una onda al ser emitida o recibida por un objeto en movimiento <sup>7</sup>.

La energía electromagnética transmitida viaja a través del aire a aproximadamente a la velocidad de la luz de tal manera que normalmente la energía viaja línea recta por el espacio libre, a velocidad constante.

Las ondas electromagnéticas se reflejan sobre superficies conductoras; por lo tanto, si las ondas reflejadas retornan al punto de emisión de las ondas se puede inferir que se encuentra un obstáculo en la trayectoria de propagación <sup>8</sup>, como se muestra en la **Figura 1-1**.

El retardo y las características de esta señal reflejada sirven al radar para determinar la posición, velocidad e incluso propiedades morfológicas del obstáculo encontrado.



**Figura 1-1:** Funcionamiento de un Sistema Radar.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

<sup>7</sup> GOMEZ, Fernandez & SALAR, Emilio. *Tipos de Radar* [blog]. 2009. [Consulta: 15 octubre 2015]. Disponible en: <http://electromagnetismo2009.blogspot.com/2009/05/3.html>

<sup>8</sup> WOLFF, Christian. *Fundamentos Radar* [en línea]. [Consulta: 5 diciembre 2015]. Disponible en: <http://www.radartutorial.eu/01.basics/Principio%20B%C3%A1sico%20de%20Funcionamiento.es.html>

### 1.1.3. Tipos de Radares

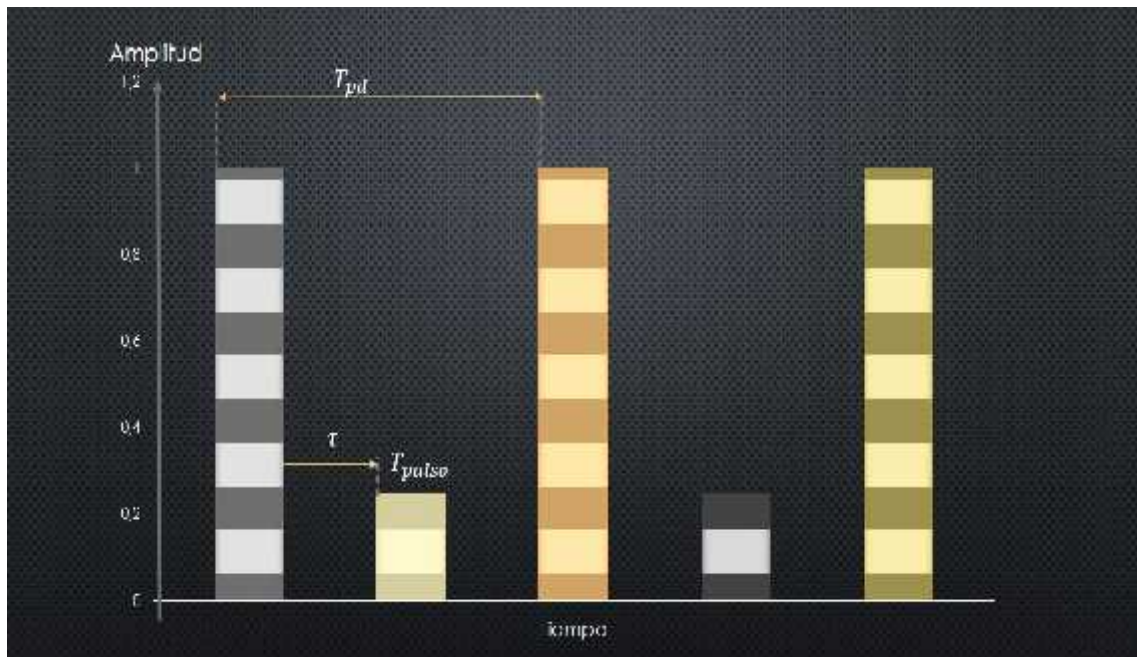
En función de la señal transmitida se clasifica en tipos de radares:

#### 1.1.3.1 Radar de Impulsos

El radar de impulsos envía señales en ráfagas muy cortas (millonésimas de segundo) pero de una potencia muy elevada. Para poder determinar la distancia el radar de pulsos mide el tiempo que la señal tarda en alcanzar el objetivo y volver al receptor (tiempo de vuelo <sup>9</sup>). Conociendo la velocidad de propagación en el medio ( $v$ ), puede ser determinada la distancia al blanco mediante la Ecuación. 1-1.

$$R = \frac{v \cdot t}{2} \quad \text{Ecuación. 1-1}$$

El máximo rango está limitado por la tasa de repetición de pulsos, el eco capturado tiene que encontrarse dentro del periodo  $T_{pd}$  entre dos pulsos, mientras que la resolución está determinada mediante el ancho del pulso  $T_{pulso}$  como se muestra en la Figura 2-1.



**Figura 2-1.** Forma Ideal de una Señal Transmitida por un Radar de Impulso.

Fuente: [www.umag.cl/investigacion/dpa/radar/sistemas/ecuaciones/figura-03.gif](http://www.umag.cl/investigacion/dpa/radar/sistemas/ecuaciones/figura-03.gif)

<sup>9</sup> GOMEZ, Fernandez & SALAR, Emilio. *Tipos de Radar* [blog]. 2009. [Consulta: 15 octubre 2015]. Disponible en: <http://electromagnetismo2009.blogspot.com/2009/05/3.html>

### 1.1.3.2 Radar de Onda Continúa

Los radares de onda continua utilizan señales continuas en vez de ráfagas cortas

- El Radar Doppler: Realiza medidas precisas de la velocidad de un objeto. Este tipo de radar transmite una onda continua de frecuencia fija. Cuando esta señal encuentra un objeto en movimiento la frecuencia de la onda reflejada cambia con respecto a la transmitida. Utilizando esta variación de frecuencia el radar determina la velocidad del objetivo.
- Los Radares FM: Emiten señales continuas pero en este caso moduladas en frecuencia. A diferencia del radar doppler, estas variaciones en frecuencia de la señal transmitida permiten no sólo conocer la velocidad del objetivo sino también su posición<sup>10</sup>.

### 1.1.4 Frecuencias de Operación

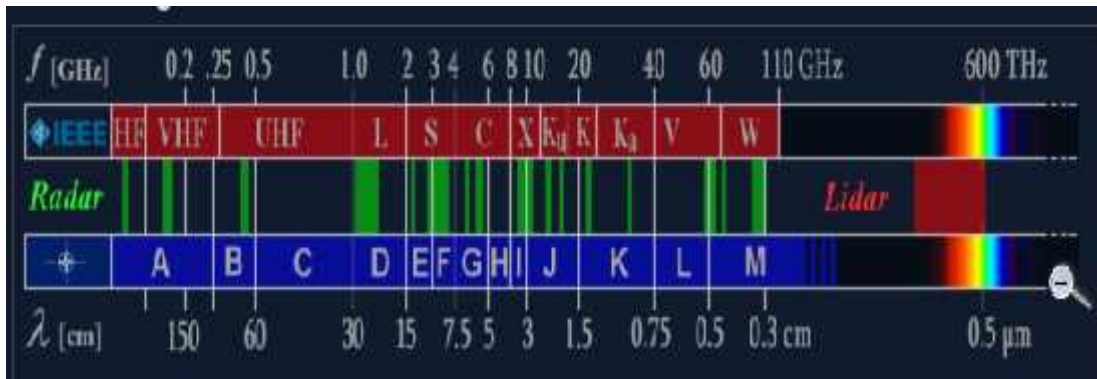
Frecuencia de funcionamiento del radar se elige sobre la base de una serie de consideraciones que son importante factores comerciales como: resolución espacial, propagación, materiales de penetración, interferencias electromagnéticas, interferencias de radiofrecuencia y limitaciones de ancho de banda.

El rango de frecuencia en la que operan los Radares es desde 220MHz a 35GHz, aunque dependiendo de la aplicación pueden trabajar en los 2MHz en radares de tierra de alta frecuencia o en radares milimétricos en los cuales trabajan en frecuencias mayores a 94GHz.

Las bandas en las que opera el Sistema Radar a lo largo del espectro electromagnético se muestran en la **Figura 3-1**. Estas bandas fueron asignadas a letras por medio del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) para que el RADAR pueda operar de manera estándar.

---

<sup>10</sup> GÓMEZ FERNÁNDEZ, Carlos & SALAR ARCUSA, Emilio. *Tipos de Radar* [blog]. [Consulta: 5 diciembre 2015] Disponible en: <http://electromagnetismo2009.blogspot.com/2009/05/3.html>



**Figura 3-1.** Frecuencia de Operación del Radar

Fuente: <http://www.radartutorial.eu/07.waves/pic/radarfrequencies.print.png>

Las frecuencias para las bandas VHF, UHF y L, se utiliza para radares de exploración y búsqueda. Seguidamente en las frecuencias L, S, C, se aplica para búsqueda y seguimiento de objetos. La banda C y X permite controlar el fuego y obtener imágenes. Finalmente en las frecuencias X, Ku, K, Ka y W, trabajan para la detección de misiles<sup>11</sup>.

Los sistemas de radar pueden trabajar en diversas frecuencias de transmisión. Pero tienen ciertas afectaciones meteorológicas los radares que operan con mayor frecuencias, entre mayor frecuencia mejor es la precisión del sistema radar. La **Figura 4-1**.muestra tipos de radares con las bandas de frecuencia utilizadas.

### 1.1.5 Radar Laser ( LADAR)

El NIST (Instituto Nacional de Estándares y Tecnología) acogió el término LADAR (Láser Detección And Ranging) para radar basados en sistemas Laser. Los sistemas LIDAR, son sistemas que utilizan una tecnología óptica de detección remota para medir propiedades de luz reflejada o esparcida hacia atrás (backscattered) por un blanco remoto, con el fin de estimar su distancia, velocidad u otras propiedades del blanco. El sistema LIDAR más utilizado es el que determina la distancia de un objeto mediante la medición del retardo de los pulsos láser<sup>12</sup>.

<sup>11</sup> **BASSEM R.,** *Radar Systems Analysis and Design Using MATLAB*, 3a ed., Alabama-USA., CRC Press., 2013., Pp. 4-8.

<sup>12</sup> **RICHMOND, Richard & CAIN, Stephen.** *Direct-Detection LADAR Systems*. 1a ed. Washington-Estados Unidos. SPIE PRESS. 2010. p. 1



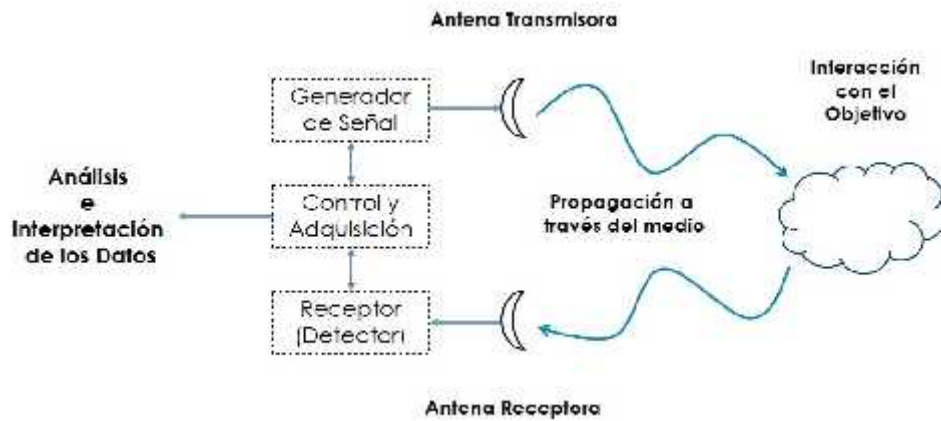
**Figura 4-1. Radares con su Banda de Operación**

**Fuente:** <http://www.radartutorial.eu/07.waves/Ondas%20y%20Rangos%20de%20Frecuencia.es.html>

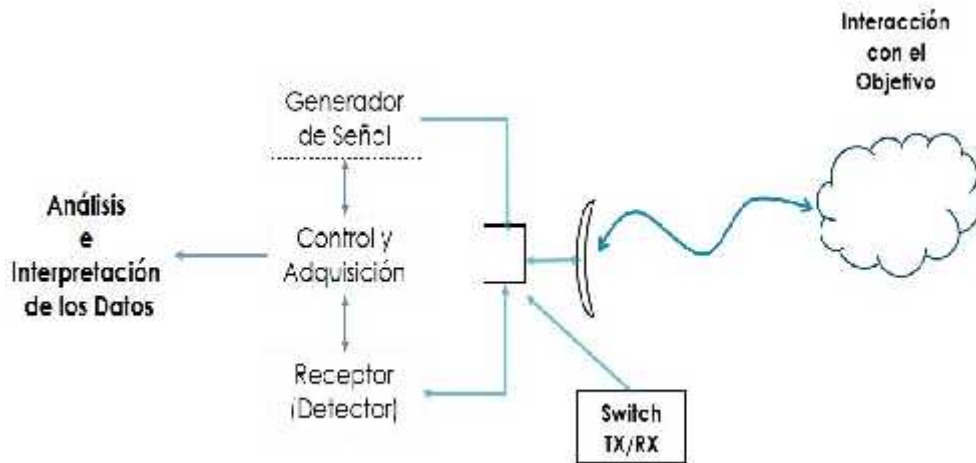
Como en un sistema radar, que usa ondas de radiofrecuencia en lugar de radiación a frecuencias ópticas, la distancia del objeto se determina midiendo el retardo entre la transmisión de un pulso y la detección de la señal reflejada<sup>13</sup>.

Todos los sistemas de detección de rango sean RADAR, LIDAR o LADAR, funcionan con la transmisión y recepción de energía electromagnética. Su diferencia entre estos sistemas es que trabajan en diferentes frecuencias. Tomando en cuenta las mismas consideraciones, como el tiempo de propagación y la teoría de la antena, aplicados en estos sistemas. Existen dos tipos de configuración en el Sistema RADAR/LIDAR como se muestra en la **Figura 5-1 y Figura 6-1**.

<sup>13</sup> **RICHMOND, Richard & CAIN, Stephen.** Direct-Selection LADAR Systems. 1a ed. Washington-Estados Unidos. SPIE PRESS. 2010. p. 1



**Figure 5-1.** Configuración Bi-estática en el Sistema LADAR/RADAR  
Fuente: RICHMOND, Richard & CAIN, Stephen. 2010. p. 2



**Figure 6-1.** Configuración Mono-estática en el Sistema LADAR/RADAR  
Fuente: RICHMOND, Richard & CAIN, Stephen. 2010. p. 2

La diferencia fundamental entre LIDAR y Radar es que en LIDAR se usan longitudes de onda mucho más pequeñas que en Radar, típicamente el ultravioleta, el visible o el infrarrojo cercano. En general, es posible medir objetos que tengan un tamaño comparable o mayor a la longitud de onda de emisión, por lo que un sistema LIDAR es en general mucho más sensible en la medición de gases y aerosoles que un radar<sup>14</sup>.

<sup>14</sup> MONTILLA et al. *Observatorio Lidar-Cefop* [en línea]. Chile. [Consulta: 20 noviembre 2015]. Disponible en: [http://www.tigo.cl/documents/sirgas/Poster\\_Lidar\\_Observatorio-LIDAR-CEFOP.pdf](http://www.tigo.cl/documents/sirgas/Poster_Lidar_Observatorio-LIDAR-CEFOP.pdf)

### **1.1.6 Reconstrucción 3D**

La Reconstrucción 3D permite reproducir un objeto real en la memoria de un computador, conservando las características físicas como es la dimensión, volumen y forma. Existen muchos métodos que permiten la reconstrucción de objetos en 3D, el objetivo de estas técnicas es crear una conexión de puntos representativos del objeto y representarla mediante un algoritmo. Estos algoritmos permiten la reconstrucción de objetos pero se diferencia en costo y calidad del mallado.

### **1.1.7 Adquisición de Datos**

Mediante la obtención de datos se busca transformar un modelo físico a una forma digital, para aquello se realiza la lectura de coordenadas de determinados puntos pertenecientes a la superficie de un objeto. Dichos datos se obtienen con máquinas de medición espaciales y escáneres, para luego ser representados en forma de x, y, z con un sistema de coordenadas locales llamada nube de puntos. La nube de puntos se asemeja a un conjunto de curvas que pertenece a la superficie de un objeto, luego son modelados los datos de manera manual para obtener el objeto tridimensional de forma digital.

Para la adquisición de datos se analiza el método a utilizar de acuerdo a la finalidad que se desea obtener, dichos métodos se clasifican si precisan de contacto o no. Los métodos de contacto son más precisos aunque lentos para la obtención de datos, al contrario, los métodos que no requieren contactos son más precisos.

Dentro de los métodos que no precisan contacto están:

- Activas: emite y recibe radiación (ultrasonidos, rayos X o luz) para la obtención de información.
- Pasivas: no emite radiación, al contrario detecta radiación ambiental.

El objetivo de esta investigación está relacionado con la adquisición de datos mediante métodos que no precisan contacto por lo que se procede a estudiar dichos métodos <sup>15</sup>.

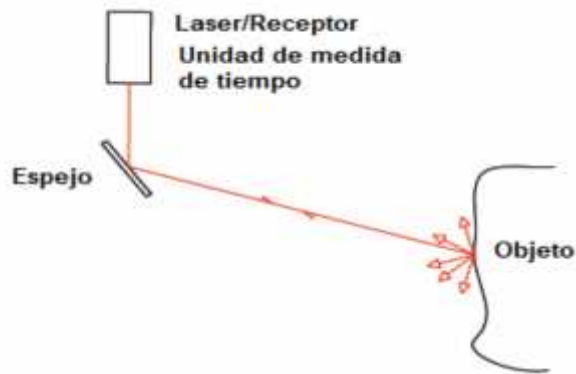
---

<sup>15</sup> NUÑO, Simón. Reconocimiento de Objetos mediante Sensor 3D Kinect. 2012. pp. 7.



### 1.1.7.1 Tiempo de Vuelo 3D o Telemetría

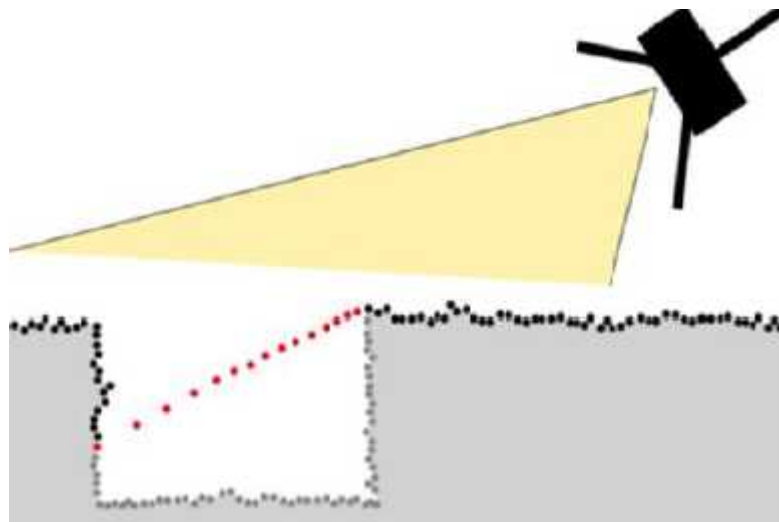
Este principio trabaja midiendo el tiempo de vuelo de un pulso láser, es decir, el tiempo que demora en ir y regresar el pulso que fue enviado y rebota en la superficie del objeto<sup>16</sup>, como se muestra en la **Figura 7-1**.



**Figura 7-1.** Principio de Tiempo de Vuelo

Fuente: QUINTANA, Matías. 2014. p. 3

El láser es emitido hacia el objeto y reflejado por el mismo, el láser rebotado es capturado por el receptor determinando de esta manera el tiempo que el láser se demoró en regresar. Durante a obtención de datos es posible que ocurran errores tangenciales, obtiene los puntos rojos en lugar de los puntos grises y negros del objeto a reconstruir en 3D como se observa en la **Figura 8-1**.



**Figura 8-1.** El Error Tangencial

Fuente: QUINTANA, Matías. 2014. p. 3

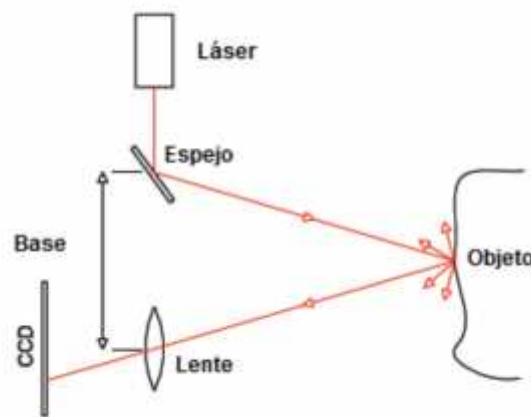
<sup>16</sup> BÖHLER, W., & MARBS, A. "3D scanning instruments". En *Proceedings of the CIPA WG 6 International Workshop on Scanning for Cultural Heritage Recording*, Ziti, Thessaloniki (pp. 9-18).



Existen escáneres de tiempo de vuelo que tienen mejor alcance, pueden obtener información mediante una cámara que emite un haz laser la cual gira 180 grados.

### 1.1.7.2 Triangulación Activa

El haz rebota en el espejo para luego incidir en el objeto, la ubicación del láser es capturado mediante una cámara CDD (Dispositivo de Carga Acoplada). Este principio es denominado Triangulación, por la forma triangular que crea el emisor, punto de incidencia del láser en el objeto y la cámara. Por medio de estos tres valores se puede determinar la distancia al objeto (ver **Figura 9-1.**)



**Figura 9-1.** Principio de Triangulación  
Fuente: QUINTANA, Matías. 2014. p. 4

Utilizan una línea de luz láser la cual puede ser horizontal o vertical, la adquisición de datos es más rápida debido que realiza un barrido al objeto de manera perpendicular en lugar de un solo punto a la vez, aunque existe zonas oscuras las cuales son formas irregulares del objeto siendo invisibles durante el recorrido del láser.

Este método permite obtener información del color de cada punto basándose en matrices con referencias RGB (rojo, verde, azul) de manera que se genera la relación entre la superficie y los atributos de color, dando como resultado la visualización de la textura del modelo<sup>17</sup>. La precisión del modelo del ángulo del vértice opuesto al escáner, a medida que se aleja del 90 grados menor es la precisión, limitando de esta manera el tamaño de la escena.

---

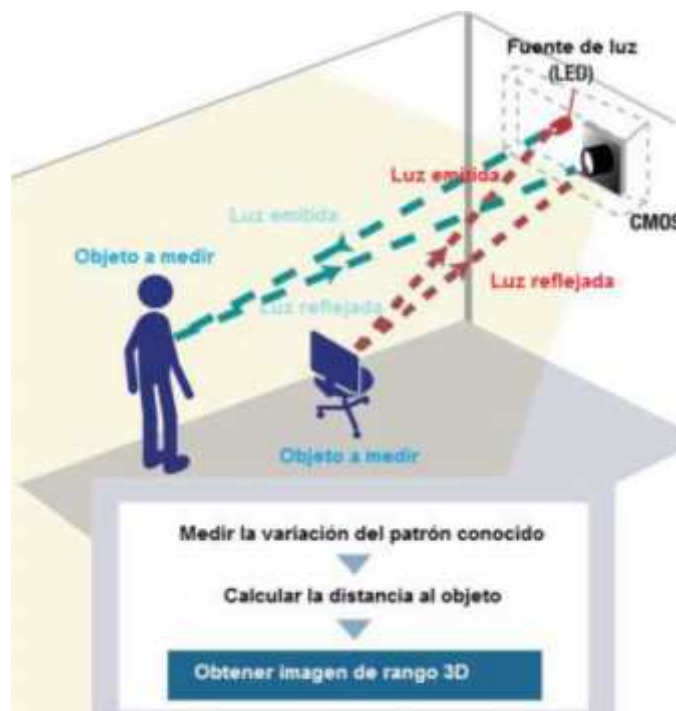
<sup>17</sup> GUAMAN GUALPA, Klever & SOSA GUZMAN, Andrés. *Diseño e Implementación de un Sistema para Digitalización en 3D mediante el uso de Matlab.*, 2013, p:11.

### 1.1.7.3 Luz Estructurada

Este principio es similar al de tiempo de vuelo, se diferencian por que la distancia del objeto se obtiene de acuerdo a la deformación del patrón el cual puede ser bandas de frecuencias o luz codificada. El modelo puede ser unidimensional o bidimensional, en el caso de unidimensional es una línea la cual se proyecta sobre el objeto con un láser, mientras que el modelo bidimensional es una cuadrícula, donde se considera una serie de líneas verticales paralelas de laser que barren horizontalmente al objeto.

Tiene como ventaja la velocidad, debido a que escanea múltiples puntos en lugar de uno a la vez, logrando analizar todo el campo entero inmediatamente. Las cámaras de profundidad que utilizan el método de luz estructurada son las cámaras RGB-D, un ejemplo es el dispositivo Kinect de Microsoft.

Existe numerosa documentación acerca de la adquisición de datos a través de la Kinect. En el artículo publicado por Cruz<sup>18</sup>. Dicho sensor ha permitido llevar acabo muchos avances. El patrón de luz estructurada del Kinect se puede apreciar en la **Figura 10-1**.



**Figura 10-1.** Principio de Funcionamiento del Escáner de Luz Estructurada.

Fuente: QUINTANA, Matías. 2014. p. 4

<sup>18</sup> CRUZ et al. "Kinect and rgbd images: Challenges and applications". *SIBGRAPI Tutorial*. ( 2012), (Brazil)

### 1.1.8 Análisis del Sistema Radar

Existen muchos parámetros que se toma en cuenta para iniciar el desarrollo del sistema radar.

#### 1.1.8.1 Resolución del Sistema

Es la mínima distancia entre dos blancos de similares características que el sistema de radar es capaz de determinar. Para obtener la resolución se toma en cuenta parámetros como el ancho de banda ( $B$ ), exactitud de la distancia ( $e_d$ ), la velocidad de la luz ( $c$ ), frecuencia máxima ( $f_{\max}$ ) y de muestreo ( $fs$ ) como se muestra en la Ecuación. 2-1 y Ecuación. 3-1.

$$fs = 2 * f_{\max} = B \quad \text{Ecuación. 2-1}$$

Resolución del sistema

$$R = e_d = \frac{c}{B} = \frac{c}{2 * f_{\max}} \quad \text{Ecuación. 3-1}$$

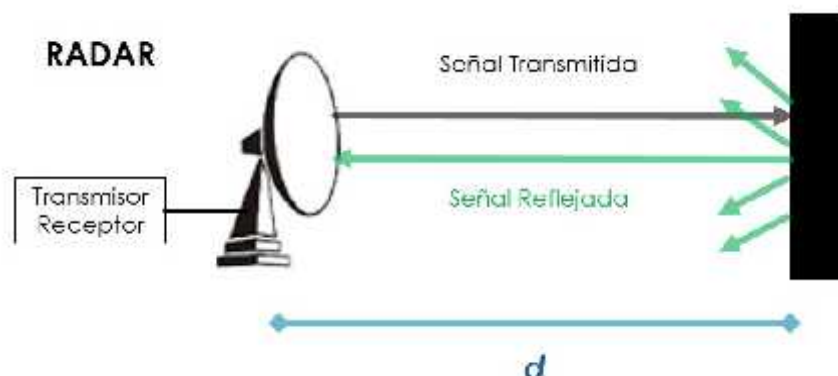
#### 1.1.8.2 Distancia Máxima de Análisis de Señal

La máxima distancia que la señal recibida del sistema radar es analizada depende del tiempo final como se muestra en la Ecuación. 4-1(ver Figura 11-1.)

$$d_{\max \text{teorica}} = \frac{c * t_f}{2} \quad \text{Ecuación. 4-1}$$

Aunque en objetivos prácticos esta distancia máxima es reducida a la mitad ( Ecuación. 5-1).

$$d_{\max \text{practica}} = \frac{c * t_f}{4} \quad \text{Ecuación. 5-1}$$



**Figura 11-1.** Señal Transmitida – Reflejada

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 1.1.9 Detección del Objeto

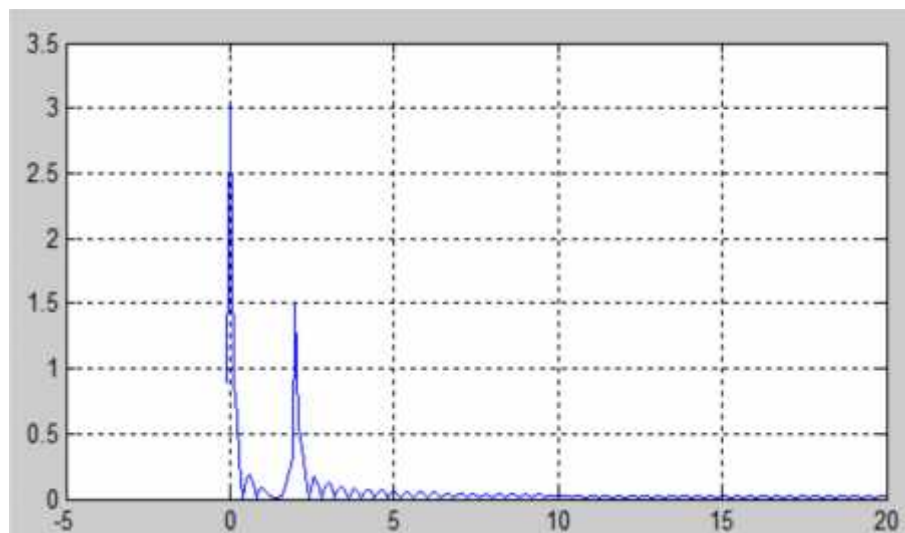
La detección del objeto se genera desde la señal transmitida la misma que es modulada partiendo de una función coseno la cual es procesada a través de ecuaciones. Dicha onda está en función del vector de tiempo y de la frecuencia de muestreo, donde la velocidad de propagación ( $c$ ) es la velocidad de la luz ( $c$ ), siendo  $f$  la frecuencia de la señal transmitida,  $t$  el tiempo y  $\phi$  es el ángulo de desfase (ver **Ecuación. 6 1**).

$$V(t) = c \cos(2\pi f t + \phi) \quad \text{Ecuación. 6 1}$$

La onda transmitida al ser reflejada por el objeto genera un retardo de tiempo y un desfase, donde el retardo es generado durante el tiempo de ida y de regreso de la onda (ver **Ecuación. 7 1**).

$$\text{Ecuación. 7 1}$$

La multiplicación de las dos señales muestra si existe algún objeto detectado, para esto es necesario de la utilización de la Transformada Rápida de Fourier, la misma que permite pasar del dominio del tiempo al dominio de la frecuencia. Los picos representan la detección de un objeto como se muestra en la **Figura 12-1**.



**Figura 12-1.** Transformada Rápida De Fourier

Fuente: ÁLVAREZ, René & SHAGÑAY, Sandra. 2015. p. 32

### 1.1.10 Distancia del Objeto Detectado

Para obtener la distancia a la que se encuentra el objeto se analiza la señal Chirp la cual cambia de frecuencia desde una baja a un alta o viceversa en un mismo periodo de tiempo determinado. La señal Chirp Transmitida está dada por una amplitud  $K$ , frecuencia angular  $\omega_0$  y fase constante  $\phi_0$ , donde  $f(t)$  es la función de la frecuencia que cambia en el tiempo como se muestra en la **Ecuación. 8-1**.

$$V(t) = V_0 \cos(2\pi f(t)t + \phi_0) \quad \text{Ecuación. 8-1}$$

La frecuencia variante en el tiempo  $f(t)$  depende de la rapidez con que cambia la frecuencia  $k$ , donde se toma en cuenta la frecuencia de inicio de la onda  $f_0$  y la frecuencia final  $f_1$  en una duración  $T$  (ver **Ecuación. 9-1** y **Ecuación. 10-1**).

$$f(t) = \frac{k}{2}t + f_0 \quad \text{Ecuación. 9-1}$$

$$k = \frac{(f_1 - f_0)}{T} \quad \text{Ecuación. 10-1}$$

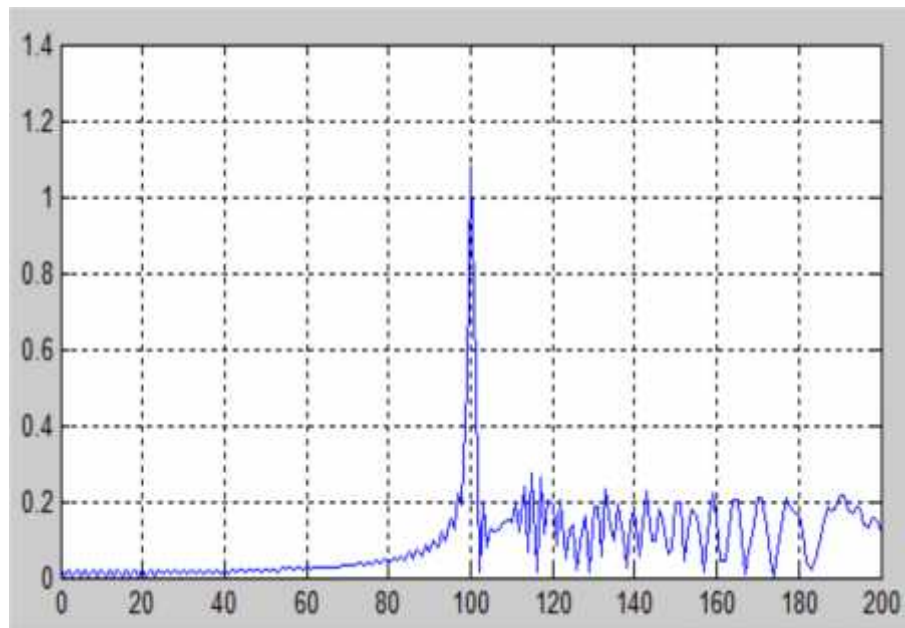
Existe un retardo en el tiempo entre la señal Chirp Recibida y la Transmitida, dicha señal depende del retardo de tiempo y un desfase como se observa en la **Ecuación. 11-1**.

$$V_r(t) = V_0 K \cos\left(2\pi\left(\frac{k}{2}(t - 2t_r) + f_0\right)(t - 2t_r) + \phi_0\right)\mu(t - 2t_r) \quad \text{Ecuación. 11-1}$$

La multiplicación de las dos señales chirp permite mostrar la posición del objeto detectado (ver **Ecuación. 12-1**), para esto es necesario de la utilización de la Transformada Rápida de Fourier, la misma que permite pasar del dominio del tiempo al dominio de la frecuencia.

$$V_f(t) = V_0 \cos\left(2\pi f(t)t + \phi_0\right) \cdot V_0 K \cos\left(2\pi\left(\frac{k}{2}(t - 2t_r) + f_0\right)(t - 2t_r) + \phi_0\right)\mu(t - 2t_r) \quad \text{Ecuación. 12-1}$$

El pico más alto (ver **Figura 13-1**.) representa la distancia a la que se encuentra el objeto analizado.



**Figura 13-1.** Transformada Rápida de Fourier

Fuente: ÁLVAREZ, René & SHAGÑAY, Sandra. 2015. p. 37

## 1.2 Kinect

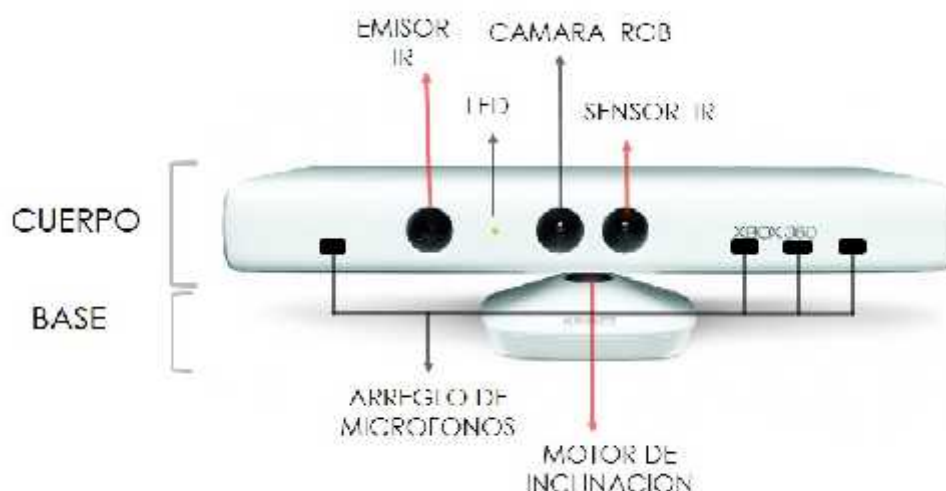
Es un sistema creado por la empresa Microsoft que permite a los usuarios interactuar con la consola de videojuegos Xbox 360 sin tener contacto físico con un controlador. Fue lanzado en noviembre del 2010. Posee los siguientes componentes: cámara RGB, micrófono de múltiples matrices, sensor de profundidad, motor de inclinación y Chip PrimeSense que permite ejecutar el software para el reconocimiento de gestos, objetos e imágenes, comandos de voz y obtener la profundidad de los objetos<sup>19</sup>.

Además posee un adaptador USB para conectar a un ordenador y un adaptador de corriente para la fuente de alimentación externa (ver **Figura 14-1.**)

### 1.2.1 Cámara RGB

Posee una entrada de 3 canales, permite detectar los colores rojo, verde y azul, con una capacidad de resolución de hasta 1280x960 en la captura de imágenes. La cámara RGB es similar a una cámara digital, la luz atraviesa el lente y es dirigida a un filtro encargado de separar los colores primarios, los cuales son proyectados sobre un sensor fotosensible.

<sup>19</sup> HOIEM, D. "Computational Photography". *University of Illinois*. (2010)



**Figura 14-1.** Componentes del Kinect.

Fuente: ABHIJIT, Jana. 2012. p. 9

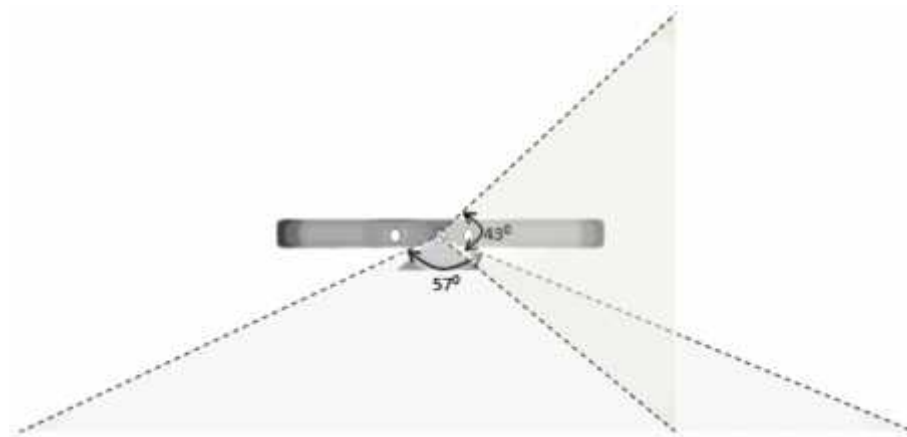
Se genera una señal eléctrica mediante dicho sensor en función de la intensidad de la señal que incide sobre dicho sensor, esta señal es convertida por medio de un ADC (Analog Digital Convert), para luego ser analizada y reconstruida mediante la interpolación que permite rellenar espacios donde falta información<sup>20</sup>.

Esta cámara detecta las variaciones de intensidad mas no distingue los colores, la señal de datos devueltos por el Kinect es una sucesión de cuadros de imágenes fijas, para obtener una imagen en color se descompone la imagen en los colores primarios, los cuales son proyectados sobre distintas zonas del sensor, la cámara se encarga de detectar la cantidad de intensidad de cada uno por separado.

La distancia del usuario para obtener mejor resolución es de 1,2 a 4 metros, el rango de profundidad es 400 mm a 8000 mm, funcionando correctamente a temperatura de 5 a 35 grados centígrados. Posee un ángulo de visión horizontal de 57 grados y vertical de 43 grados como se muestra en la **Figura 15-1**<sup>21</sup>.

<sup>20</sup> HOIEM, D. "Computational Photography". *University of Illinois*. (2010)

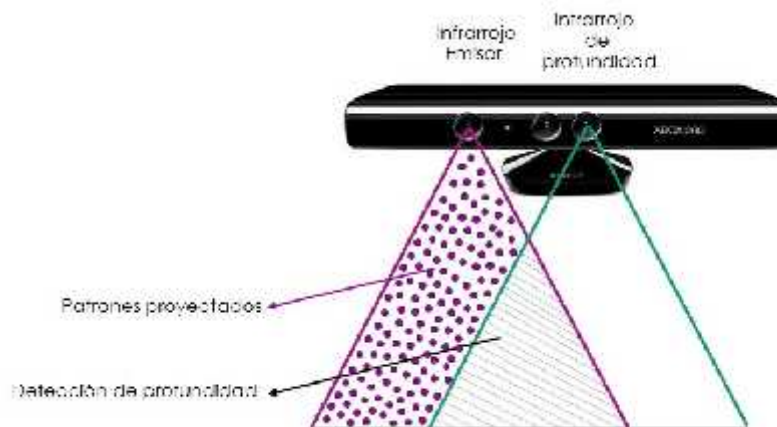
<sup>21</sup> ABHIJIT, Jana. Kinect for Windows SDK Programming Guide. 1a ed. Birmingham – Mumbai. Packt Publishing Ltd. 2012. pp. 8-9



**Figura 15-1.** Ángulos de Limitación.  
Fuente: ABHIJIT, Jana. 2012. p. 10

### 1.2.2 Sensor de Profundidad

Está formado por dos componentes: un emisor infrarrojo y un sensor CMOS, ambos se encuentran alineados en el eje x del Kinect. La disposición en el Kinect se basa en un método similar al de triangulación activa entre el emisor y la cámara. Con la diferencia que el método que utiliza el Kinect se trata de un proceso de dos fases, uno de calibración y otra de funcionamiento, el cual es denominado “Luz Estructurada” como muestra en la **Figura 16-1**<sup>22</sup>.



**Figura 16-1.** Método de Luz Estructurada  
Fuente: ABHIJIT, Jana. 2012. p. 11

<sup>22</sup> ABHIJIT, Jana. Kinect for Windows SDK Programming Guide. 1a ed. Birmingham – Mumbai. Packt Publishing Ltd. 2012. pp. 10-11



En la fase de calibración el emisor proyecta un patrón de puntos no visibles para el ojo humano sobre un plano variando su distancia, son reflejados al chocar en el objeto siendo grabados por la cámara IR receptora y luego son comparados con un patrón conocido (ver **Figura 17-1.**).



**Figura 17-1.** Patrón de Puntos Proyectado  
Fuente: NUÑO, Javier. 2015. p. 53

En la fase de funcionamiento analiza el patrón de puntos proyectados y obtiene la profundidad mediante un cuadro de píxeles. El sensor de profundidad captura 30 FPS (cuadros por segundo) con una resolución VGA de 640x480 píxeles. Se pueden obtener hasta 2048 niveles de profundidad posible.

El rango de profundidad del sensor está entre 0,4 y 4 metros. El sensor de profundidad tiene un campo angular de 60° en el sentido horizontal y 43° en el sentido vertical. La mínima distancia que se puede utilizar el sensor es aproximadamente 0,7m dando a lugar a un campo visual de 0,87m en el sentido horizontal y 0,63m en el sentido vertical.

### 1.2.3 Matriz de Micrófono

Contienen cuatro micrófonos que capturan la información de audio, cada canal procesado en 16 bits de audio con un rango de frecuencia de 16KHz. No solo captura el sonido sino que también permite localizar la dirección de la onda de audio, lo que lo convierte en un micrófono bidireccional al identificar la fuente del sonido y reconocerla voz independientemente del ruido y el eco presente en el ambiente (ver **Figura 18-1.**)<sup>23</sup>.

---

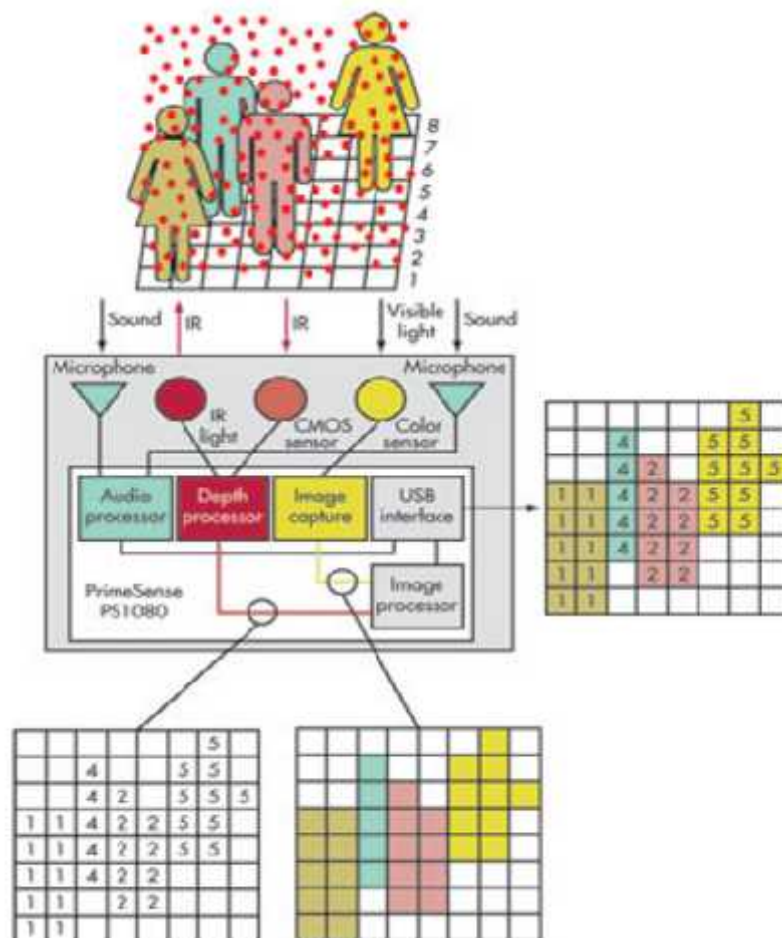
<sup>23</sup> ABHIJIT, Jana. Kinect for Windows SDK Programming Guide. 1a ed. Birmingham – Mumbai. Packt Publishing Ltd. 2012. pp. 9, 121-155, 211-236



**Figura 18-1.** Matriz de Micrófono  
Fuente: ABHIJIT, Jana. 2012. p. 14

#### 1.2.4 Chip PrimeSense PS1080

El Chip PrimeSense, gestiona la información de audio y visual de forma independiente. Proyectando un patrón de puntos de IR del sensor los cuales son detectados mediante el sensor CMOS. Dicho patrón es modificado de acuerdo a los objetos que reflejan la luz. El Chip toma los resultados identificando los puntos y realizando la traducción de su estado a un valor de profundidad generando un mapa de profundidad, como se muestra en la **Figura 19-1**.

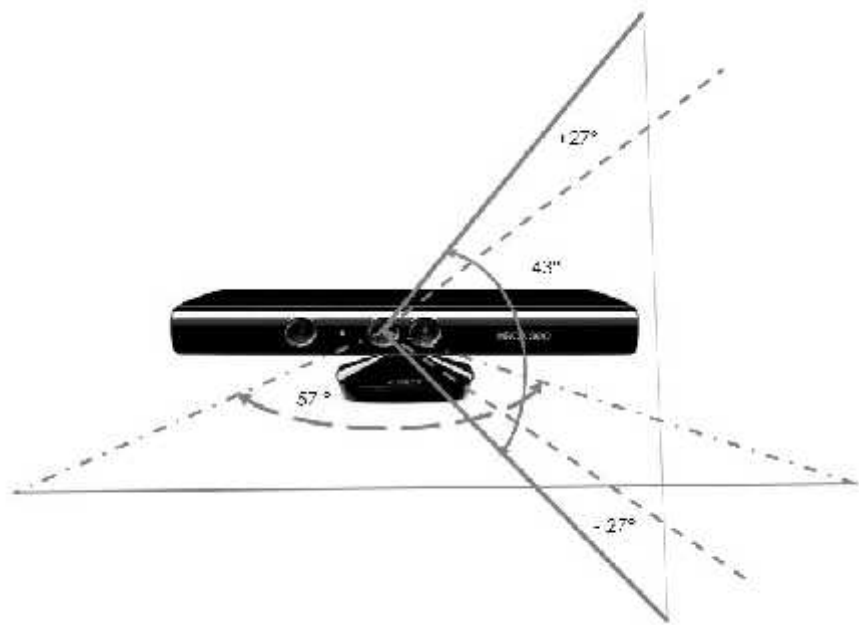


**Figura 19-1.** Proceso de Adquisición de Datos de la Cámara RGB y del Sensor de Profundidad

Fuente: <http://electronicdesign.com/embedded/how-microsoft-s-primesense-based-kinect-really-works>

### 1.2.5 Motor de Inclinación

Permite ajustar la inclinación, para lo cual posee tres engranajes que permiten al sensor inclinar verticalmente 27 grados, lo que indica que el Kinect puede desplazarse 27 grados hacia arriba y hacia abajo como se muestra en la **Figura 20-1**.



**Figura 20-1.** Ángulo del Motor de Inclinación

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 1.2.6 Especificaciones Técnicas

Las principales características de cada componente que conforma el dispositivo Kinect se muestran a continuación en la **Tabla 1-1**.

### 1.2.7 Controladores y Plataformas

Los controladores para el sensor Kinect pueden funcionar en MacOSX10.7, Windows 7 y Ubuntu 11.10, están disponibles en distintas licencias: OpenKinect, OpenNI y el Kinect Windows. En la **Tabla 2-1** se muestra las ventajas y desventajas entre los controladores <sup>24</sup>.

---

<sup>24</sup> CHUYA et al. Diseño e Implementación de un Sistema para el Análisis del Movimiento Humano usando Sensores Kinect. 2013. pp. 15

**Tabla 1-1.** Características del Kinect

KINECT XBOX 360		
1	Sensores de Profundidad	Cámara: 640×480 CMOS, 16 bits, 30 FPS.
		Infrarrojo: resolución 0,01 eje Z; -0,003m ejes X e Y.
2	Cámara RGB	VGA 640×480; 32 bits; 30FPS.
3	Micrófonos	Rango: $\pm 50^\circ$ .
		Cancelación ruido: 20dB.
		Resolución: 24 bits PCM.
		Frecuencia: 16kHz.
4	Motor	Ángulo Inclinación: $\pm 27^\circ$ .
5	Acelerómetro	2G/4G/8G para un rango de 2G con una precisión de $\pm 1$
6	Rango de Profundidad	Min: 0,4m a 1,2m
		Max: 3,5m a 4m
7	Ángulo de Visión	Horizontal: $57^\circ$
		Vertical: $43^\circ$

Fuente: CHUYA et al. 2013. p. 14.

### 1.2.7.1 Controlador OpenKinect

Este controlador es de licencia libre, los controladores desarrollados permiten acceder directamente a la cámara permitiendo obtener imágenes RGB y de profundidad. Las librerías del controlador están avaladas por la comunidad de software libre. Este sistema requiere de un tiempo de calibración en el que el sujeto a monitorizar debe tener una postura inicial definida (ver **Figura 21-1.**)

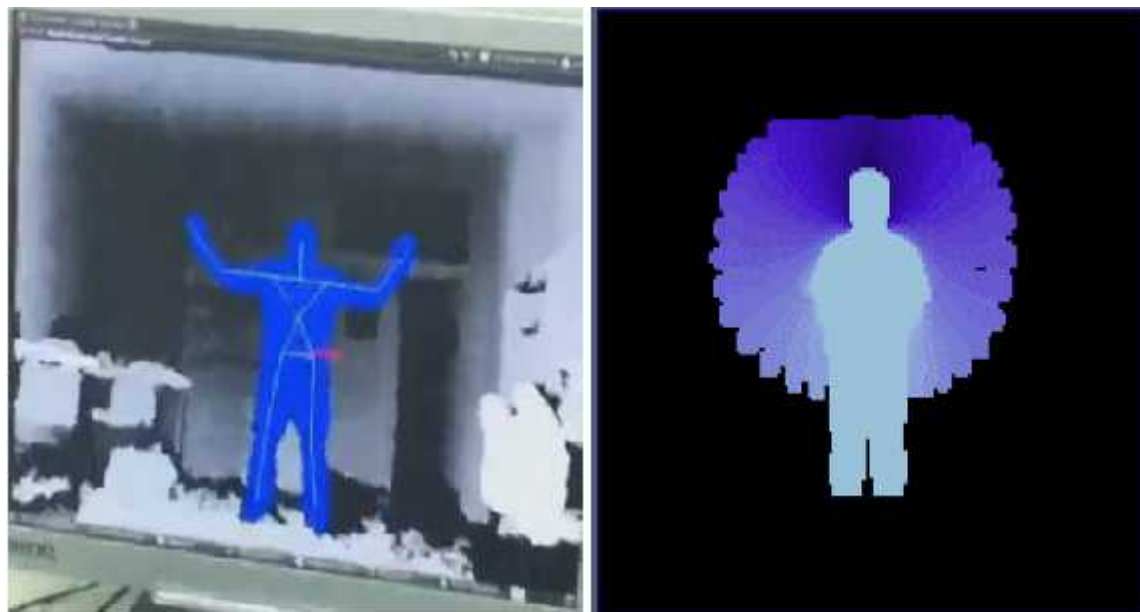
### 1.2.7.2 Controlador OpenNI/NITE

Permite obtener datos de profundidad, imágenes RGB y audio, fue desarrollado por PrimeSenseOpenNI se maneja bajo licencias de código libre y comercial. NITE tiene licencia comercial y permite acceder a la esquelización del usuario para el reconocimiento de posturas

y gestos <sup>25</sup>. Para la obtención de un esqueleto 3D se ancla las articulaciones del cuerpo en 24 puntos, de esta manera se obtiene información de las coordenadas y la orientación de los puntos (ver **Figura 22-1**) <sup>26</sup>.



**Figura 21-1.** Captura de Profundidad mediante Controlador OpenKinect.  
Fuente: MARTÍNEZ et al. 2011. p. 3



**Figura 22-2.** Esqueletización y Seguimiento de una Persona mediante OpenNI.  
Fuente: MARTÍNEZ et al. 2011. p. 3

---

<sup>25</sup> **CHUYA et al.** Diseño e Implementación de un Sistema para el Análisis del Movimiento Humano usando Sensores Kinect. 2013. pp. 15-16

<sup>26</sup> **POPPE, R.** "Vision-based human motion analysis: An overview". Computer Vision and Image Understanding. 2007, pp. 4-18.

### 1.2.7.3 Controlador Microsoft Kinect SDK

Este controlador tiene similitud con el OpenNI/NITE con la diferencia que permite el seguimiento de hasta dos usuarios y reconoce a seis en un área aproximada de 10 metros cuadrados (ver **Figura 23-1**). Fue lanzado por Microsoft y está sujeto a licencias de uso libre de Microsoft y sus políticas comerciales.



**Figura 23-1.** Reconocimiento de Personas mediante Microsoft Kinect SDK

**Fuente:** <https://blogs.msdn.microsoft.com/kinectforwindows/2014/07/15/the-kinect-for-windows-v2-sensor-and-free-sdk-2-0-public-preview-are-here/>

**Tabla 2-1.** Ventajas y Desventajas entre el Controlador OPENNI y Microsoft

Controladores del Dispositivo Kinect		
	SDK MICROSOFT	SDK OPENNI/NITE
Ventajas	Soporte por Audio.	Uso libre y comercial.
	Incluye las Manos, Pies y Clavícula.	Adquiere datos de las manos por seguimiento y reconocimiento de gestos.
	Seguimiento de Cuerpo completo.	Seguimiento de cuerpo entero.
	No necesita postura de calibración.	Puede calibrar la profundidad y el color de la imagen.
	Soporte para el motor de inclinación.	Calculo de la rotación de la articulación.
	Mejor tratamiento de articulaciones no visibles.	Plataforma múltiple: Windows XP, Vista, 7, Linux y MacOSX.
	Soporta múltiples sensores.	Soporte incorporado para grabación y reproducción.
	Instalación simple.	
	Gran cantidad de información disponible.	
	Sistema de reconocimiento de gestos.	
	La versión 1.7 permite correr al Kinect SDK sobre máquinas virtuales.	
Desventajas	Licencia única para uso no comercial.	Instalación Complicada.
	Mayor consumo de recursos computacionales.	Sin soporte para audio y motor de inclinación.
	Solamente para Windows 7 y 8.	Necesita de una postura de calibración.
	Sin soporte incorporado para grabación y reproducción.	Articulaciones no visibles no son estimadas.
	Sin soporte para transmitir los datos sin procesar el sensor infrarrojo.	Compatibles con varios sensores, aunque la configuración y enumeración es complicada.

Fuente: CHUYA et al. 2013. p. 15.

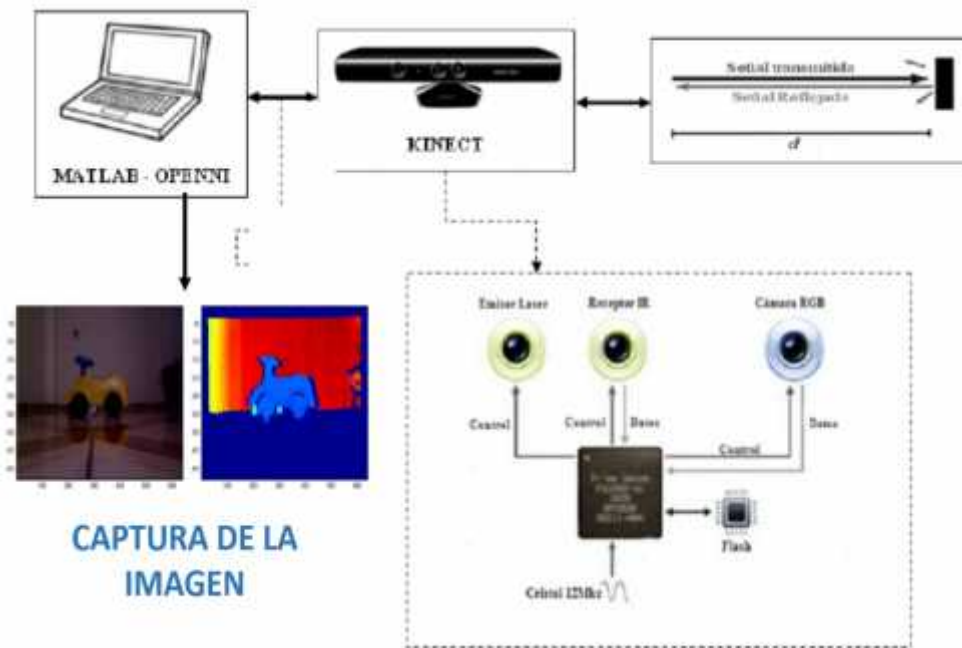


## CAPITULO II

### 2. SISTEMA RADAR PARA RECONSTRUCCION DE OBJETOS 3D MEDIANTE EL DISPOSITIVO KINECT.

#### 2.1 Sistema Radar.

Para elaborar el Sistema Radar que permita la reconstrucción de objetos 3D se adaptó la conexión del dispositivo Kinect a un computador y por medio del software Open NI y Matlab sea posible detectar objetos obteniendo un esquema del Sistema Radar como el que se muestra en la **Figura 1-2**.



**Figura 1-2.** Sistema Radar utilizando el Dispositivo Kinect.

Fuente: ÁLVAREZ, René & SHAGÑAY, Sandra. 2015. p. 49

#### 2.1.1 Funcionamiento Sistema Radar.

El sensor Kinect es la herramienta fundamental de Sistema Radar debido a su funcionalidad, en el documento de tesis “Sistema Radar para Análisis y Reconstrucción de Perfiles Topográficos a Media Escala” (Álvarez R., Shagñay S., 2015, p: 49-54) se detalla el funcionamiento del sensor Kinect destacando que la proyección de luz infrarroja transmitida incide sobre los objetos a



través de un patrón de puntos por tal motivo la señal rebota contra el objeto y es detectada por el receptor infrarrojo CMOS donde posteriormente por medio de la programación en el software Matlab se depura la información obtenida para estimar la distancia en un punto central a la que el objeto se encuentra del Dispositivo Kinect.

### **2.1.2 Librerías.**

Para el funcionamiento del Sistema Radar mediante el dispositivo Kinect se utiliza librerías tales como Open NI, primesense NITE y los archivos .Mex para enlazar el dispositivo Kinect con el software Matlab, en el **Anexo A** se muestra los instaladores necesarios para enlazar al Kinect con la PC.

La empresa Primesense que en la actualidad es de propiedad de APPLE desarrolló los controladores y el software que permite la extracción de información del dispositivo Kinect al computador de manera que la librería Open NI provee una infraestructura genérica basada en APIs con código abierto para servir a la cámara, sensor y audio del Kinect como controlador y poder acceder a los dispositivos que incluye el Kinect.

Además Primesense desarrolló la librería NITE cuyo código no es abierto pero sirve para acceder a funciones avanzadas del Kinect como es el seguimiento del cuerpo en tiempo real.

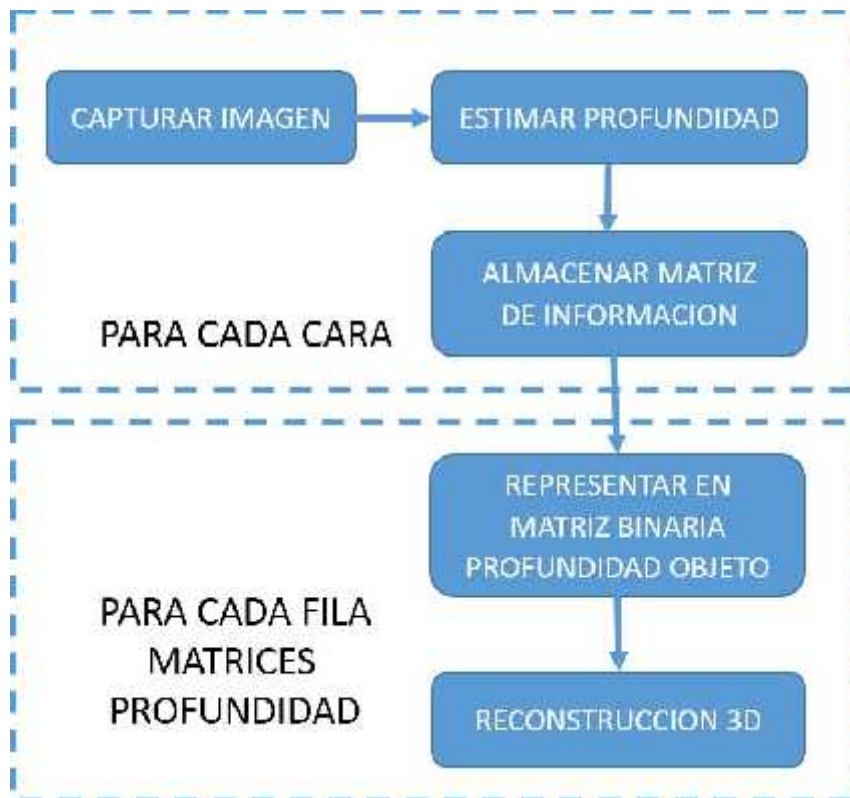
Para interactuar con el Kinect y el software Matlab es necesario utilizar los archivos .Mex que fueron desarrollado por Sourceforge.net como proyecto de investigación teniendo la ventaja de ser archivos de código abierto útiles para el software Matlab, Estos archivos se lo denomina Kinect-Mex y permiten obtener información necesaria de las imágenes de RGB e Infrarrojo capturadas por el Kinect en entorno Matlab.

## **2.2 Código Matlab.**

La código Matlab que se utiliza en el programa que permite la Reconstrucción de Objetos en forma Tridimensional está definido para primero extraer las mediciones de profundidad para las cuatro caras laterales que constituyen el área cúbica ficticia donde se encuentra el Objeto por tal motivo se almacena una matriz con información de profundidad para la primera cara, segunda

cara, tercera cara y cuarta cara lateral para posteriormente a los datos de las matrices representar en matrices binarias para permitir visualizar la Reconstrucción del Objeto en 3D.

A continuación la **Figura 2-2** muestra el Diagrama de Flujo con las etapas que tiene el código Matlab para la reconstrucción de Objetos.



**Figura 2-2.** Diagrama de Flujo Etapas de Código Matlab.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 2.2.1 Enlace Matlab con Sensor Kinect.

Para enlazar el Kinect y el programa Matlab se añade la ruta de la carpeta donde se encuentra localizados los archivos .Mex, cuyos archivos permiten prender y apagar el dispositivo Kinect en Matlab además tiene los archivos con diferentes modos de funcionamiento.

```
addpath('./Mex');
```

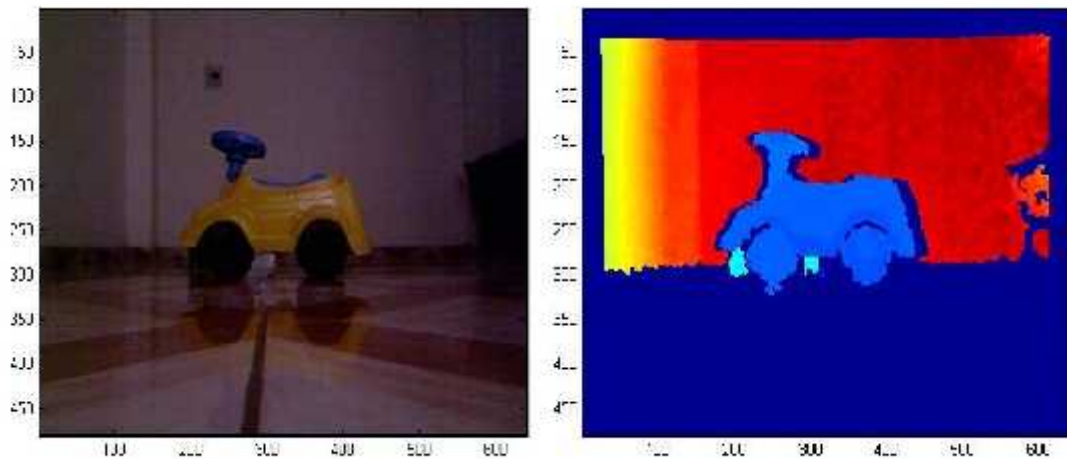
Para conectar el Kinect con Matlab en modo de operación SamplesConfig.xml, se utiliza la función “mxNiCreateContex”. Este modo de funcionamiento se guarda en Matlab como contexto y permite obtener tanto imágenes RGB como de profundidad a una resolución de 640 x

480 píxeles (ver **Figura 3-2**), esta estructura se mantendrá durante toda la ejecución del programa.

```
Context = mxNiCreateContext('Config/SamplesConfig.xml');
```

Se debe alinear la información de la imagen de color con la imagen de profundidad debido a que la cámara Infrarroja no se encuentra en la misma posición que la cámara RGB, por lo cual las imágenes están desplazadas la una de la otra. La información de la imagen RGB se guarda en la variable `rgb` y la información de la imagen de profundidad en la variable `depth`.

```
option.adjust_view_point = true;  
[rgb, depth] = mxNiImage(context);
```



**Figura 3-2.** Imagen RGB e Imagen de Profundidad.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

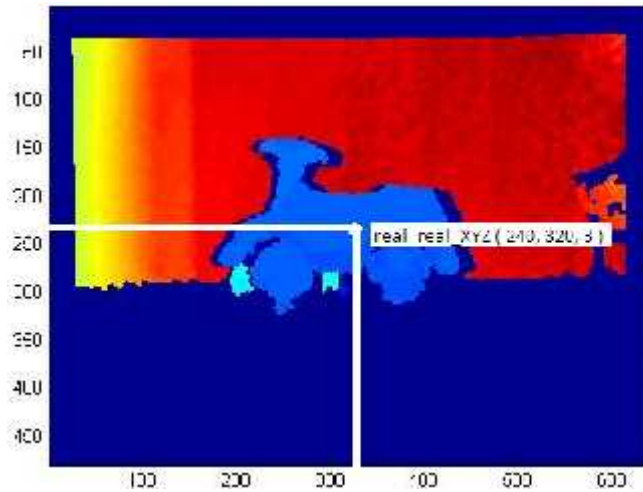
### 2.2.2 Selección de Información.

La información obtenida por el Kinect por defecto se representa en píxeles ( $u, v$ ) por esta razón se debe transformar la información de la imagen de profundidad a unidades de longitud [mm].

```
real_XYZ = mxNiConvertProjectiveToRealWorld(context, depth);
```

Para obtener la distancia del objeto con respecto al Kinect se selecciona la información de la imagen de profundidad en un punto específico, para esto se añaden las coordenadas que seleccionaran la ubicación de los datos en la matriz `real_XYZ` siendo de interés los datos centrales, como se muestra en la **Figura 4-2**.

real\_XYZ ( 240, 320, 3 )

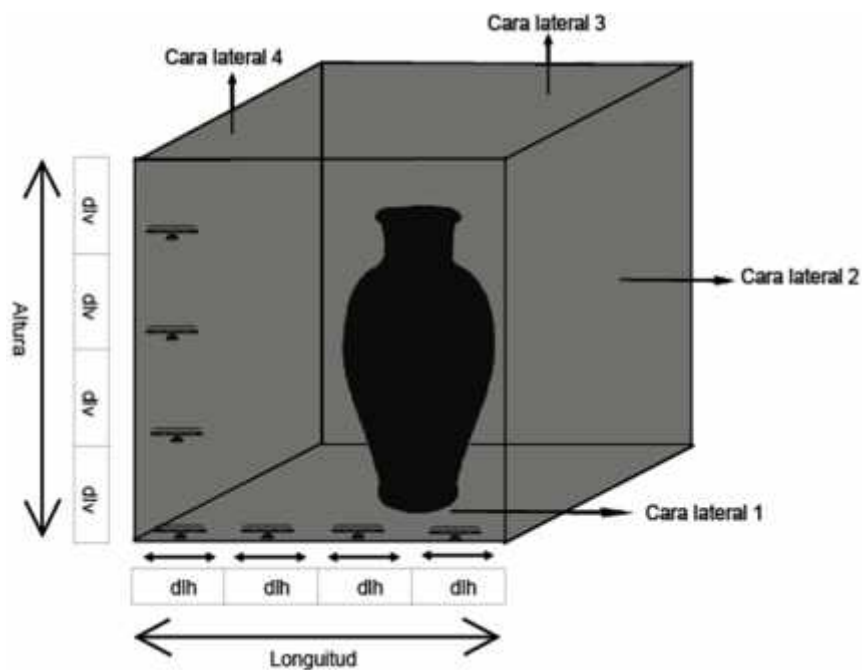


**Figura 4-2.** Obtención de Información.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 2.2.3 Almacenamiento de Información.

Para almacenar la información de cada cara lateral del área cúbica ficticia que cubre al objeto se define los parámetros tales como: Longitud de la arista horizontal, Altura del Objeto, Distancia a la que se tomara cada medición horizontalmente y la distancia que se tomará cada medición verticalmente como se observa en la **Figura 5-2**.



**Figura 5-2.** Parámetros para la Reconstrucción 3D.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

La información que guarda la variable `real_XYZ` se utiliza cada vez que se realice la medición de distancia para los desplazamientos horizontales y desplazamientos verticales, para calcular la distancia que existe entre el Kinect y el Objeto se realiza la sustracción entre la variable `real_XYZ (240, 320, 3)` y la Longitud de arista horizontal, por otra parte si la Longitud de arista es menor que el valor de variable `real_XYZ(240,320,3)` será automáticamente su respuesta cero debido a que se asume entonces que no se encuentra ningún objeto para esa ubicación.

Para almacenar las mediciones realizadas de una de las caras laterales en un vector, se utiliza una estructura de repetición para repetir el proceso hasta cubrir toda la cara lateral.

```

i=0;
base=0;
alturaob=0;
intervalo=h/dl; % Numero de datos por fila
intervaloalt=alt/dlv; % Numero de filas
contalt=1;
cont=1;

while (base<= intervalo && alturaob <= intervaloalt)
    i=i+1;
    real= (real_XYZ(240,320,3)/1000); %Dato de Profundidad

    if (real<=h && real~=0 )
        y(i)=h-real; %Dato de distancia
    else
        y(i)=0;
    end

    cont=cont+1;
    base=cont;

    if (base>intervalo && alturaob<intervaloalt)
        cont=1;
        l=dl;
        base=0;
        contalt=contalt+1;
        alturaob=contalt;
    end
end

```

Una vez almacenado las mediciones en el vector, se convierte el vector en matriz para ello utilizamos las expresiones matemáticas expuestas en la **Ecuación. 1–2**, **Ecuación. 2 – 2** y **Ecuación. 3 – 2** para determinar las dimensiones de la matriz.

columnas=(h/dl); %Numero de columnas de la Matriz

t=vec2mat (y, columnas); % Matriz de la información Primera Cara.

- **Expresión Matemática.**

$$n = \left( \frac{\text{Longitud Arista Horizontal (h)}}{\text{Desplazamiento Horizontal (dl)}} \times \frac{\text{Altura (alt)}}{\text{Desplazamiento Vertical (dlv)}} \right) \quad \text{Ecuación. 1–2}$$

$$m = \frac{\text{Altura (alt)}}{\text{Desplazamiento Vertical (dlv)}} \quad \text{Ecuación. 2–2}$$

$$c = \frac{\text{Longitud Arista Horizontal (h)}}{\text{Desplazamiento Horizontal (dl)}} \quad \text{Ecuación. 3–2}$$

Donde, n representa el Número total de datos en Vector, m= Numero de filas en la matriz, c Numero de columnas de la matriz, **A** el vector almacenado y **M** la matriz de información.

**Nota:** Desplazamiento horizontal debe ser múltiplo de Longitud de arista Horizontal y Desplazamiento vertical debe ser múltiplo de Altura.

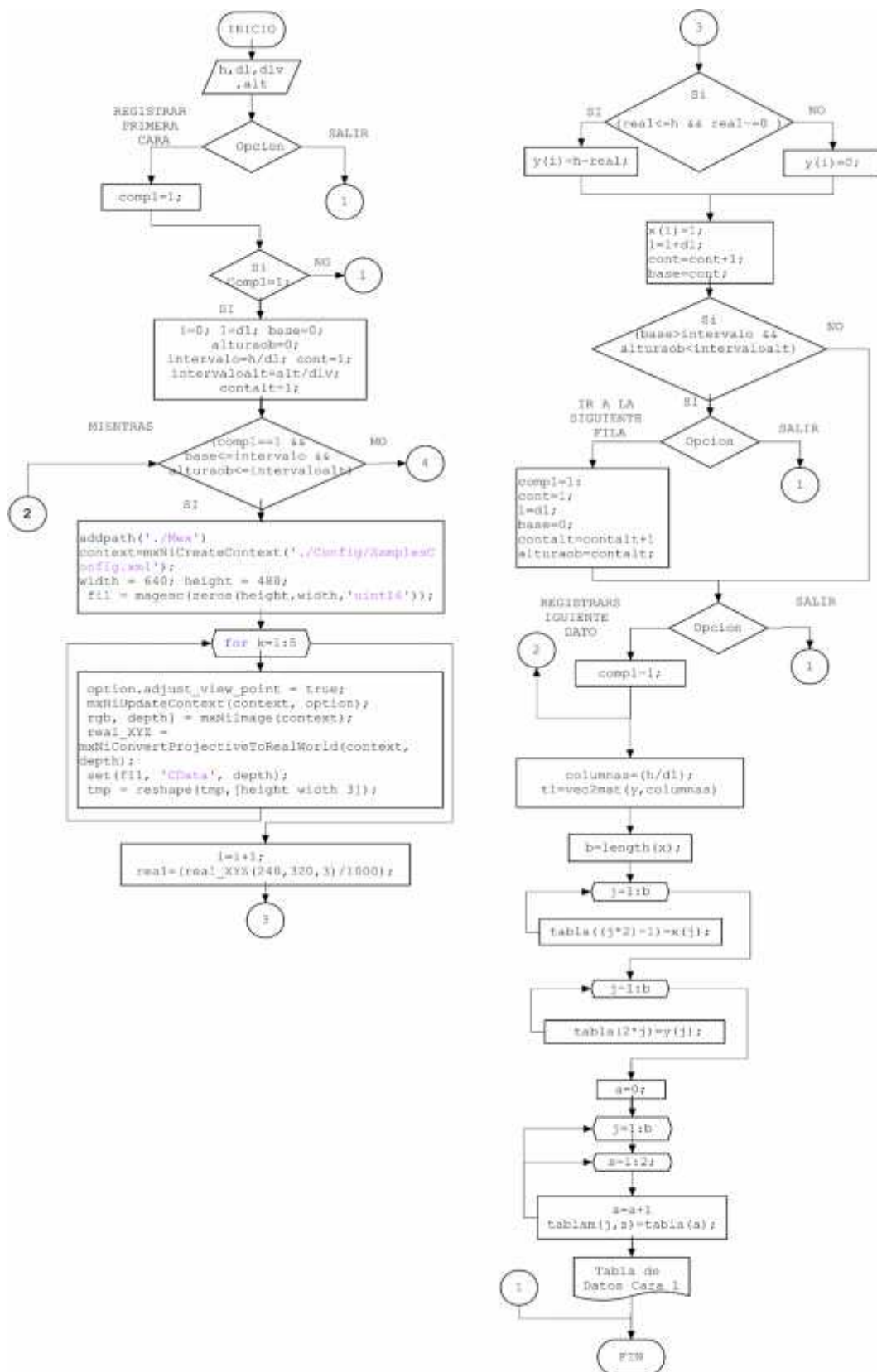
**Vector:**

$$\mathbf{A} = (a_1 \ a_2 \ a_3 \ a_4 \ . \ . \ . \ a_n)$$

**Matriz:**

$$\mathbf{M} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,c} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,c} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,c} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,c} \end{pmatrix}$$

A continuación en la **Figura 6-2**, se observa el flujograma para almacenar la matriz de información de una cara lateral.



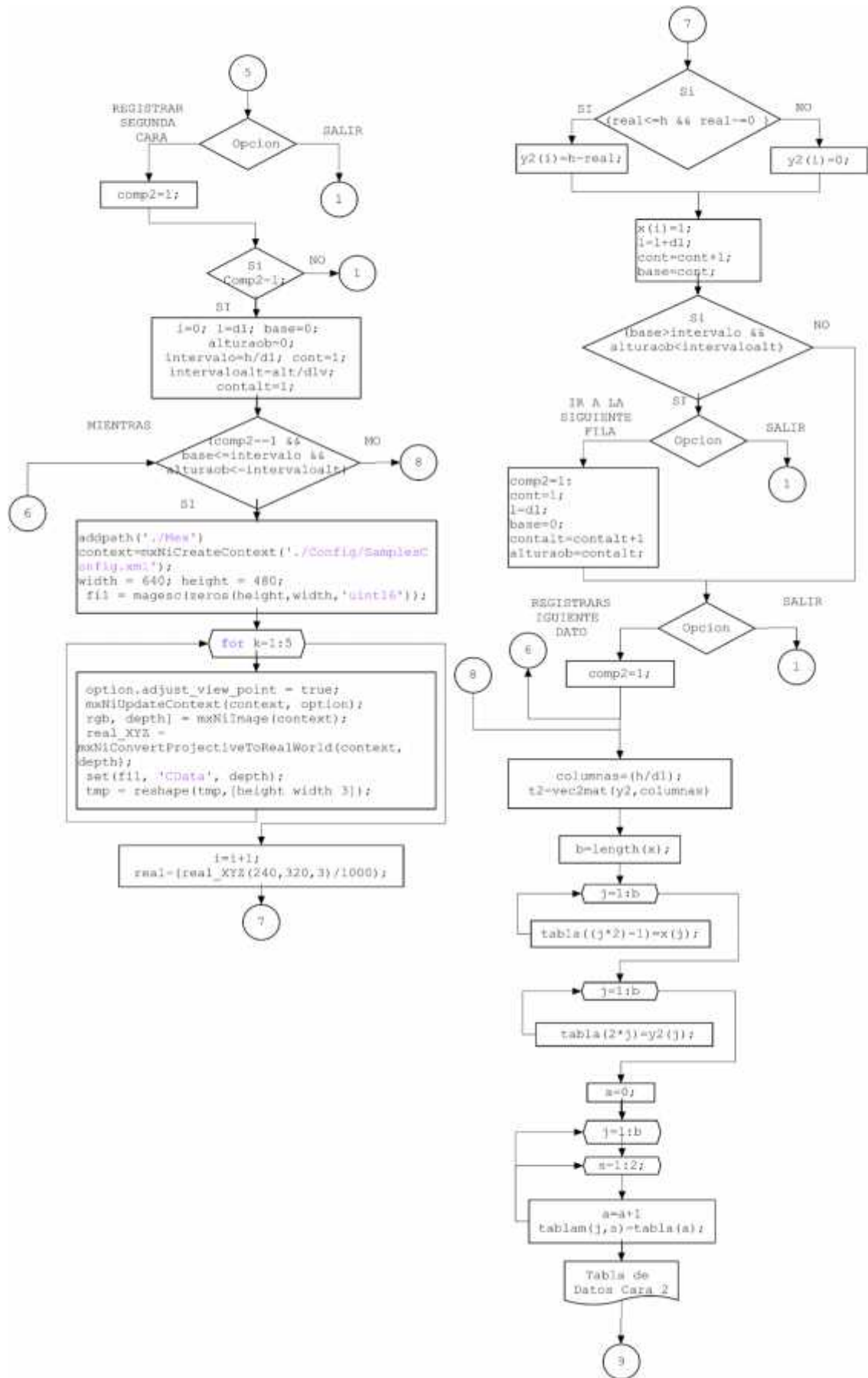
**Figura 6-2.** Flujograma para Almacenar Matriz de Información.

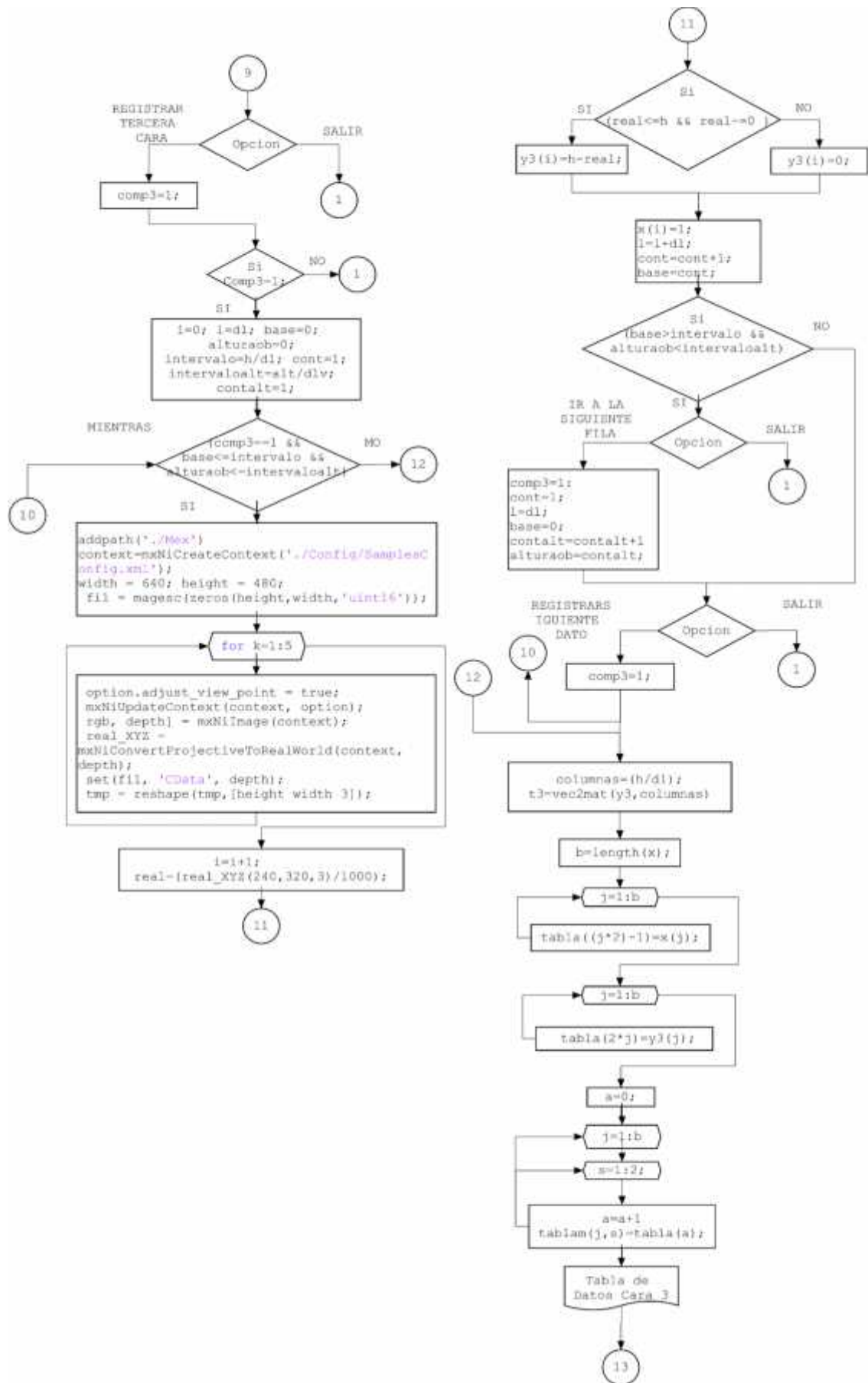
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

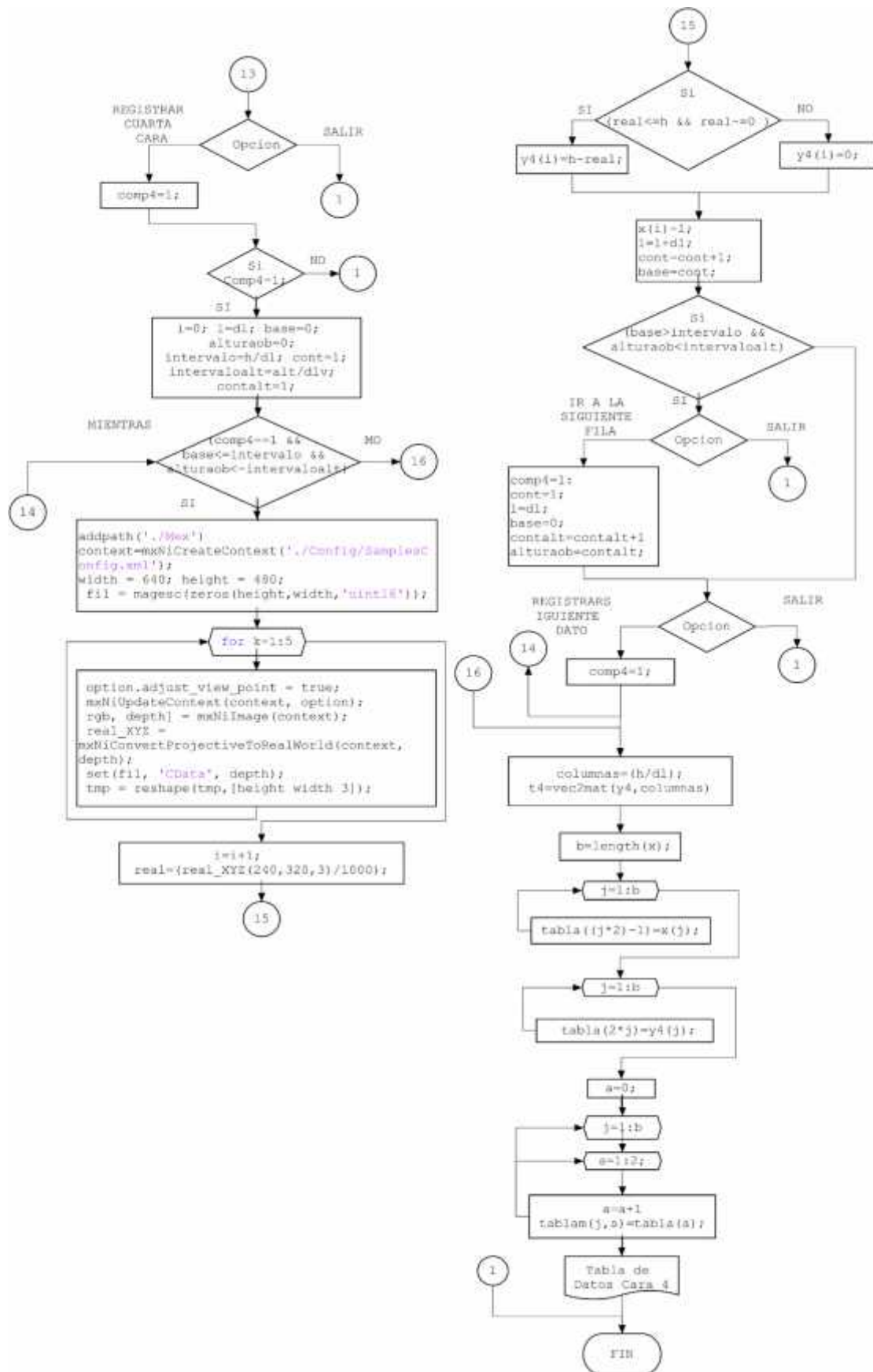
A continuación en la **Figura 7-2**, se observa el flujograma para almacenar las matrices de las caras laterales.











**Figura 7-2.** Flujograma para Almacenar Matrices de Información.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

## 2.2.4 Reconstrucción del Objeto en 3D

### 2.2.4.1 Creación de Matrices 0's

Una vez que se obtiene las matrices que guardan las medidas de profundidad de las cuatro caras laterales, se crea cuatro matrices cuadradas de 0's con dimensión igual al número de columnas que tiene una de las matrices almacenadas (Ver **Ecuación. 3–2**), las matrices 0's servirá para relacionar cada dato de una fila de la matriz de profundidad con una columna de la matriz 0's de manera que se asigna 1's en cada columna de la matriz 0's de acuerdo al valor que tiene el dato de la fila de la matriz de profundidad.

- Matriz Datos Cara Lateral**

$$\text{Matriz Cara1} \quad A_{(m \times c)} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,c} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,c} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,c} \end{pmatrix}$$

$$\text{Matriz Cara2} \quad B_{(m \times c)} = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & \dots & b_{1,c} \\ b_{2,1} & b_{2,2} & b_{2,3} & \dots & b_{2,c} \\ b_{3,1} & b_{3,2} & b_{3,3} & \dots & b_{3,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{m,1} & b_{m,2} & b_{m,3} & \dots & b_{m,c} \end{pmatrix}$$

$$\text{Matriz Cara3} \quad C_{(m \times c)} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,c} \\ c_{2,1} & c_{2,2} & c_{2,3} & \dots & c_{2,c} \\ c_{3,1} & c_{3,2} & c_{3,3} & \dots & c_{3,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & c_{m,3} & \dots & c_{m,c} \end{pmatrix}$$

$$\text{Matriz Cara4} \quad D_{(m \times c)} = \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \dots & d_{1,c} \\ d_{2,1} & d_{2,2} & d_{2,3} & \dots & d_{2,c} \\ d_{3,1} & d_{3,2} & d_{3,3} & \dots & d_{3,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{m,1} & d_{m,2} & d_{m,3} & \dots & d_{m,c} \end{pmatrix}$$

- Matrices Ceros:**

$$CA, CB, CC, CD_{(c \times c)} = \begin{pmatrix} 0_{1,1} & 0_{1,2} & 0_{1,3} & \dots & 0_{1,c} \\ 0_{2,1} & 0_{2,2} & 0_{2,3} & \dots & 0_{2,c} \\ 0_{3,1} & 0_{3,2} & 0_{3,3} & \dots & 0_{3,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{c,1} & 0_{c,2} & 0_{c,3} & \dots & 0_{c,c} \end{pmatrix}$$

### 2.2.4.2 Matriz Binaria

Para asignar 1's a las matrices 0's utilizamos la **Ecuación. 4-2** cuya ecuación determina la cantidad de 1's que serán reemplazados en cada columna, esta ecuación se utiliza para todos los datos una fila de las matrices almacenadas. Los 1's asignados representan la distancia a la que se encuentra el objeto de acuerdo a la ubicación a la que se tome su medida.

```

cuad=(columnas);
matcerosc1=zeros(cuad); %0's para Cara 1
matcerosc2=zeros(cuad); %0's para Cara 2
matcerosc3=zeros(cuad); %0's para Cara 3
matcerosc4=zeros(cuad); %0's para Cara 4

for i=1:columnas

    posc1= t1(contfilas,i); %Cara Lateral 1
    posc11=((posc1*cuad)/h);
    posc111=round(posc11);
    posc2= t2(contfilas,i); %Cara Lateral 2
    posc22=((posc2*cuad)/h);
    posc222=round(posc22);
    posc3=t3(contfilas,i); %Cara Lateral 3
    posc33=((posc3*cuad)/h);
    posc333=round(posc33);
    posc4= t4(contfilas,i); %Cara Lateral 4
    posc44=((posc4*cuad)/h);
    posc444=round(posc44);

    for j=1:posc111
        matcerosc1(j,i)=1;
    end
    for j=1:posc222
        matcerosc2(j,i)=1;
    end
    for j=1:posc333
        matcerosc3(j,i)=1;
    end
    for j=1:posc444
        matcerosc4(j,i)=1;
    end
end

```

- **Análisis Matemático.**

$$p = \frac{(A(a_{1,j}) \times \text{DimensiónMatrizCeros}(c))}{\text{LongitudAristaHorizontal}(h)} ; \quad p: \text{Numero Entero} \quad \text{Ecuación. 4-2}$$

Siendo  $p$  el numero de 1's que representa la distancia a la que se encuentra el objeto con respecto a la ubicación del Kinect.

- **Para A (1,j)**

$$p1 = \frac{(A(a_{1,j}) \times \text{DimensiónMatrizCeros}(c))}{\text{LongitudAristaHorizontal}(h)} \quad p1: \text{Numero Entero} \quad \text{Ecuación. 5-2}$$

$$\begin{cases} C1A_{i,j} = 1, & i \geq p1 \\ C1A_{i,j} = 0, & i < p1 \end{cases} \quad \forall i, j = 1, 2, \dots, c$$

Siendo  $p1$  la cantidad de 1's que representara la posición a la que se encuentra el objeto con respecto a la ubicación del Kinect para los datos de la matriz A (ver **Ecuación. 5-2**)

$$C1A_{(c \times c)} = \begin{pmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & \dots & 1_{1,c} \\ 1_{2,1} & 1_{2,2} & 1_{2,3} & \dots & 1_{2,c} \\ \vdots & \vdots & \vdots & & \vdots \\ 1_{(p1),1} & 1_{(p1),2} & 1_{p1,3} & \vdots & 1_{p1,c} \\ 0_{(p1+1),1} & 0_{(p1+1),2} & 0_{(p1+1),3} & \dots & 0_{(p1+1),c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{c,1} & 0_{c,2} & 0_{c,3} & \dots & 0_{c,c} \end{pmatrix}$$

- **Para B (1,j)**

$$p2 = \frac{(B(b_{1,j}) \times \text{DimensiónMatrizCeros}(c))}{\text{LongitudAristaHorizontal}(h)} \quad p2: \text{Numero Entero} \quad \text{Ecuación. 6-2}$$

$$\begin{cases} C1B_{i,j} = 1, & i \geq p2 \\ C1B_{i,j} = 0, & i < p2 \end{cases} \quad \forall i, j = 1, 2, \dots, c$$

Siendo  $p2$  la cantidad de 1's que representara la posición a la que se encuentra el objeto con respecto a la ubicación del Kinect para los datos de la matriz B (ver **Ecuación. 6-2**).

$$C1B_{(c \times c)} = \begin{pmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & \dots & 1_{1,c} \\ 1_{2,1} & 1_{2,2} & 1_{2,3} & \dots & 1_{2,c} \\ \vdots & \vdots & \vdots & & \vdots \\ 1_{(p2),1} & 1_{(p2),2} & 1_{p2,3} & \vdots & 1_{p2,c} \\ 0_{(p2+1),1} & 0_{(p2+1),2} & 0_{(p2+1),3} & \dots & 0_{(p2+1),c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{c,1} & 0_{c,2} & 0_{c,3} & \dots & 0_{c,c} \end{pmatrix}$$

- **Para C (1,j)**

$$p3 = \frac{C(c_{1,j}) \times \text{DimensiónMatrizCeros}(c)}{\text{LongitudAristaHorizontal}(h)} \quad p3: \text{Numero Entero} \quad \text{Ecuación. 7-2}$$

$$\begin{cases} C1A_{i,j} = 1, & i \geq p3 \\ C1A_{i,j} = 0, & i < p3 \end{cases} \quad \forall i, j = 1, 2, \dots, c$$

Siendo  $p3$  la cantidad de 1's que representara la posición a la que se encuentra el objeto con respecto a la ubicación del Kinect para los datos de la matriz C (ver **Ecuación. 7-2**).

$$C1C_{(c \times c)} = \begin{pmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & \dots & 1_{1,c} \\ 1_{2,1} & 1_{2,2} & 1_{2,3} & \dots & 1_{2,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1_{(p3),1} & 1_{(p3),2} & 1_{p3,3} & \vdots & 1_{p3,c} \\ 0_{(p3+1),1} & 0_{(p3+1),2} & 0_{(p3+1),3} & \dots & 0_{(p3+1),c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{c,1} & 0_{c,2} & 0_{c,3} & \dots & 0_{c,c} \end{pmatrix}$$

- **Para D (1,j)**

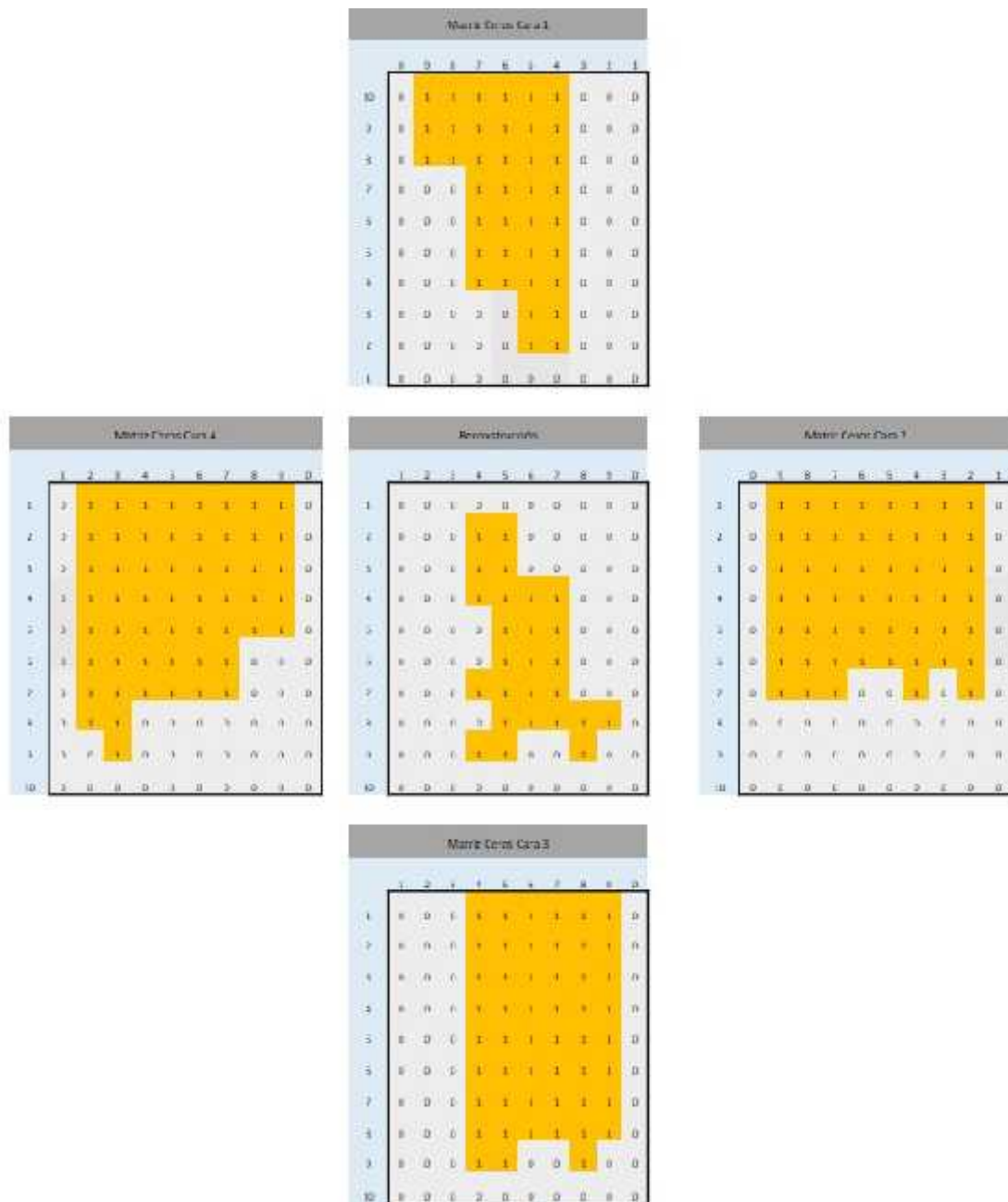
$$p4 = \frac{D(d_{1,j}) \times \text{DimensiónMatrizCeros}(c)}{\text{LongitudAristaHorizontal}(h)} \quad p4: \text{Numero Entero} \quad \text{Ecuación. 8-2}$$

$$\begin{cases} C1A_{i,j} = 1, & i \geq p4 \\ C1A_{i,j} = 0, & i < p4 \end{cases} \quad \forall i, j = 1, 2, \dots, c$$

Siendo  $p4$  la cantidad de 1's que representara la posición a la que se encuentra el objeto con respecto a la ubicación del Kinect para los datos de la matriz D (ver **Ecuación. 8-2**).

$$C1D_{(c \times c)} = \begin{pmatrix} 1_{1,1} & 1_{1,2} & 1_{1,3} & \dots & 1_{1,c} \\ 1_{2,1} & 1_{2,2} & 1_{2,3} & \dots & 1_{2,c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1_{(p4),1} & 1_{(p4),2} & 1_{p4,3} & \vdots & 1_{p4,c} \\ 0_{(p4+1),1} & 0_{(p4+1),2} & 0_{(p4+1),3} & \dots & 0_{(p4+1),c} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{c,1} & 0_{c,2} & 0_{c,3} & \dots & 0_{c,c} \end{pmatrix}$$

En la **Figura 8-2**, se observa la asignación de 1's en las matrices de 0's para los datos de una de las filas de las matrices que guardan las medidas de profundidad.

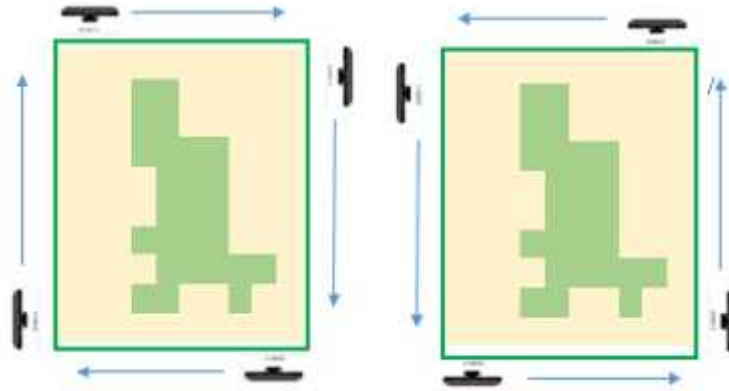


**Figura 8-2.** Asignación de 1's en Matrices O's.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

A continuación se realiza rotaciones a las matrices binarias, las rotaciones de las matrices se las realiza de acuerdo al orden que se tome la medidas de profundidad del objeto para ello se puede realizar de manera horaria o anti horaria como nos muestra la **Figura 9-2**.





**Figura 9-2.** Obtención de Datos en forma Horaria o Anti-Horaria  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 10-2** se observa la rotación de las matrices binarias para mediciones tomadas en sentido anti-horario.

```

cara1=rot90(matcerosc1,2); %Rotar 180 grados
cara2=rot90(matcerosc2,3); %Rotar 270 grados
cara3=matcerosc3;          %Rotar 0 grados
cara4=rot90(matcerosc4,1); %Rotar 90 grados

```

Matriz Ceros Cara 1	Matriz Ceros Cara 2	Matriz Ceros Cara 3	Matriz Ceros Cara 4
0 9 8 7 6 5 4 3 2 1	0 9 8 7 6 5 4 3 2 1	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0
10 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0	1 0 0 0 1 1 1 1 1 0	1 0 0 0 0 0 0 0 0 0
9 0 0 0 1 1 0 0 0 0	2 0 0 0 1 1 1 1 1 1	2 0 0 0 1 1 1 1 1 0	2 1 1 1 1 0 0 0 0 0
8 0 0 0 1 1 0 0 0 0	3 0 0 0 1 1 1 1 1 1	3 0 0 0 1 1 1 1 1 0	3 1 1 1 1 0 0 0 0 0
7 0 0 0 1 1 1 0 0 0	4 0 0 0 1 1 1 1 1 1	4 0 0 0 1 1 1 1 1 0	4 1 1 1 1 1 1 0 0 0
6 0 0 0 1 1 1 0 0 0	5 0 0 0 1 1 1 1 1 1	5 0 0 0 1 1 1 1 1 0	5 1 1 1 1 1 1 0 0 0
5 0 0 0 1 1 1 0 0 0	6 0 0 0 1 1 1 1 1 1	6 0 0 0 1 1 1 1 1 0	6 1 1 1 1 1 1 0 0 0
4 0 0 0 1 1 1 0 0 0	7 0 0 0 1 1 1 1 1 1	7 0 0 0 1 1 1 1 1 0	7 1 1 1 1 1 1 0 0 0
3 0 0 0 1 1 1 1 0 0	8 0 0 0 1 1 1 1 1 1	8 0 0 0 1 1 1 1 1 0	8 1 1 1 1 1 1 1 0 0
2 0 0 0 1 1 1 1 1 0	9 0 0 0 1 1 1 1 1 1	9 0 0 0 1 1 0 0 1 0 0	9 1 1 1 1 1 1 1 1 0
1 0 0 0 1 1 1 1 1 0	10 0 0 0 0 0 0 0 0 0	10 0 0 0 0 0 0 0 0 0	10 0 0 0 0 0 0 0 0 0

**Figura 10-2.** Rotación de Matrices Binarias.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

A continuación de las rotaciones de las matrices se procede a realizar la multiplicación (ver **Ecuación. 9–2**) posición por posición de las matrices binarias obteniendo como resultado otra matriz binaria donde cada elemento con valor 1, representa una fracción del Objeto. Además se crea una matriz binaria para la fila 1 de las matrices A, B, C, D como muestra la **Figura 11-2**.

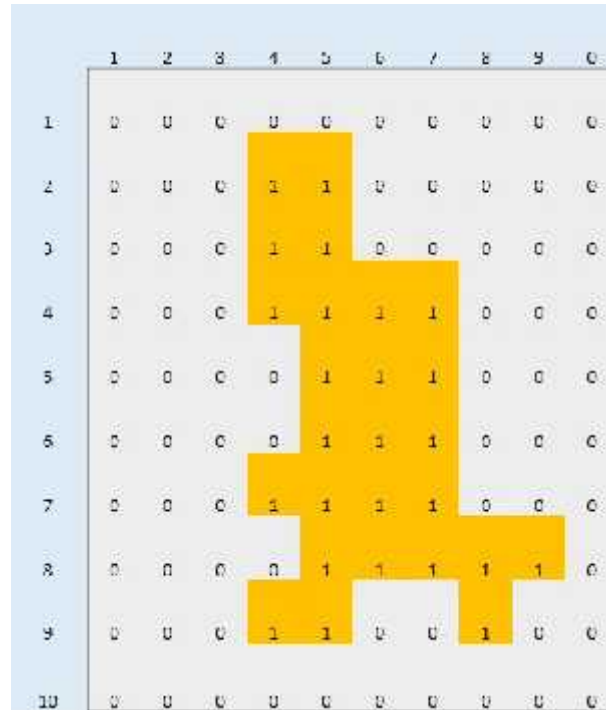
```

for i=1:cuad
    for j=1:cuad
        A(i,j)=(cara1(i,j)*cara2(i,j)*cara3(i,j)*cara4(i,j));
    end
end

```

- **Expresión matemática**

$$R(c \times c) = \text{ref}(\text{ref}(A(i, j))^T)^T \times \text{ref}(B(i, j))^T \times C(i, j) \times \left( \text{ref}(D(i, j)) \right)^T \quad \text{Ecuación. 9-2}$$



**Figura 11-2.** Matriz Binaria.

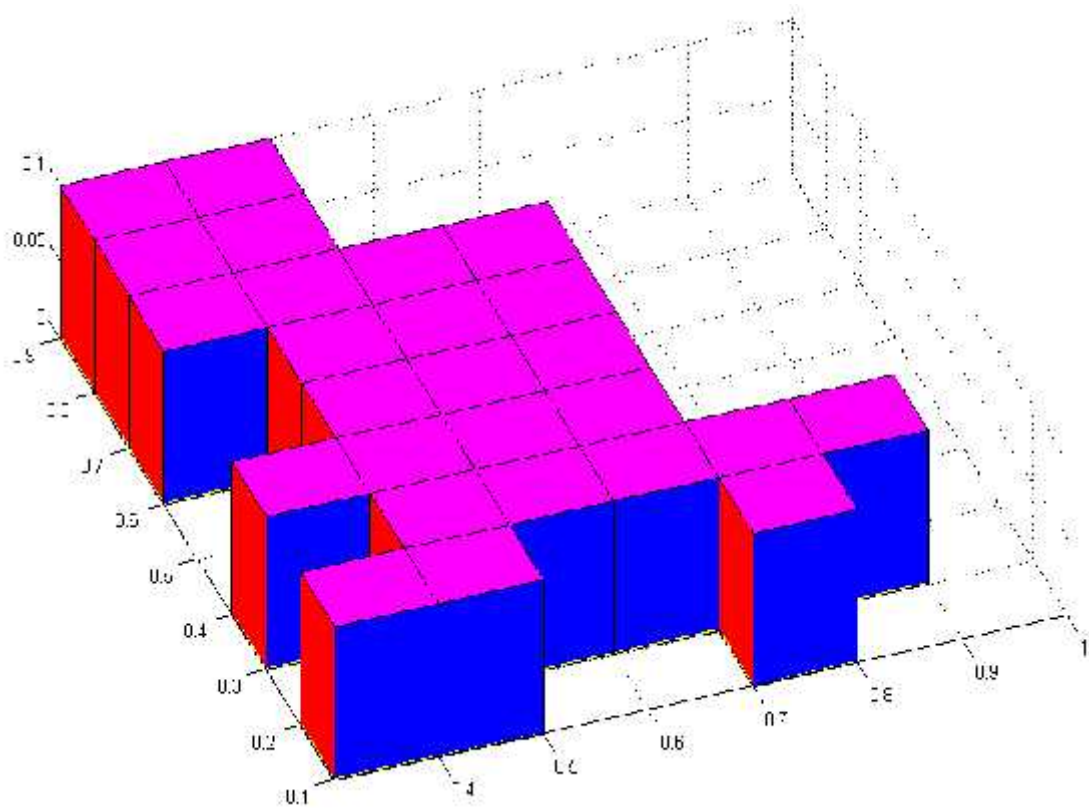
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Finalmente para poder obtener La reconstrucción del objeto en 3D se grafica la matriz binaria resultante mediante la función boxplot3 la cual permite representar los elementos con valor 1 en cubos, como muestra la **Figura 12-2**.

Para reconstruir un objeto con n filas de informacion en las matrices se debe utilizar una esturctura de repeticion que permita reutilizar las matrices 0's para asignar 1's y obtener la matriz binaria por cada fila de información de las matrices A,B,C y D (ver **Ecuación. 10-2**).

$$R_{(c \times c \times n)} = \text{ref}(\text{ref}(A(i, j))^T)^T \times \text{ref}(B(i, j))^T \times C(i, j) \times \left( \text{ref}(D(i, j)) \right)^T \quad \text{Ecuación.10-2}$$

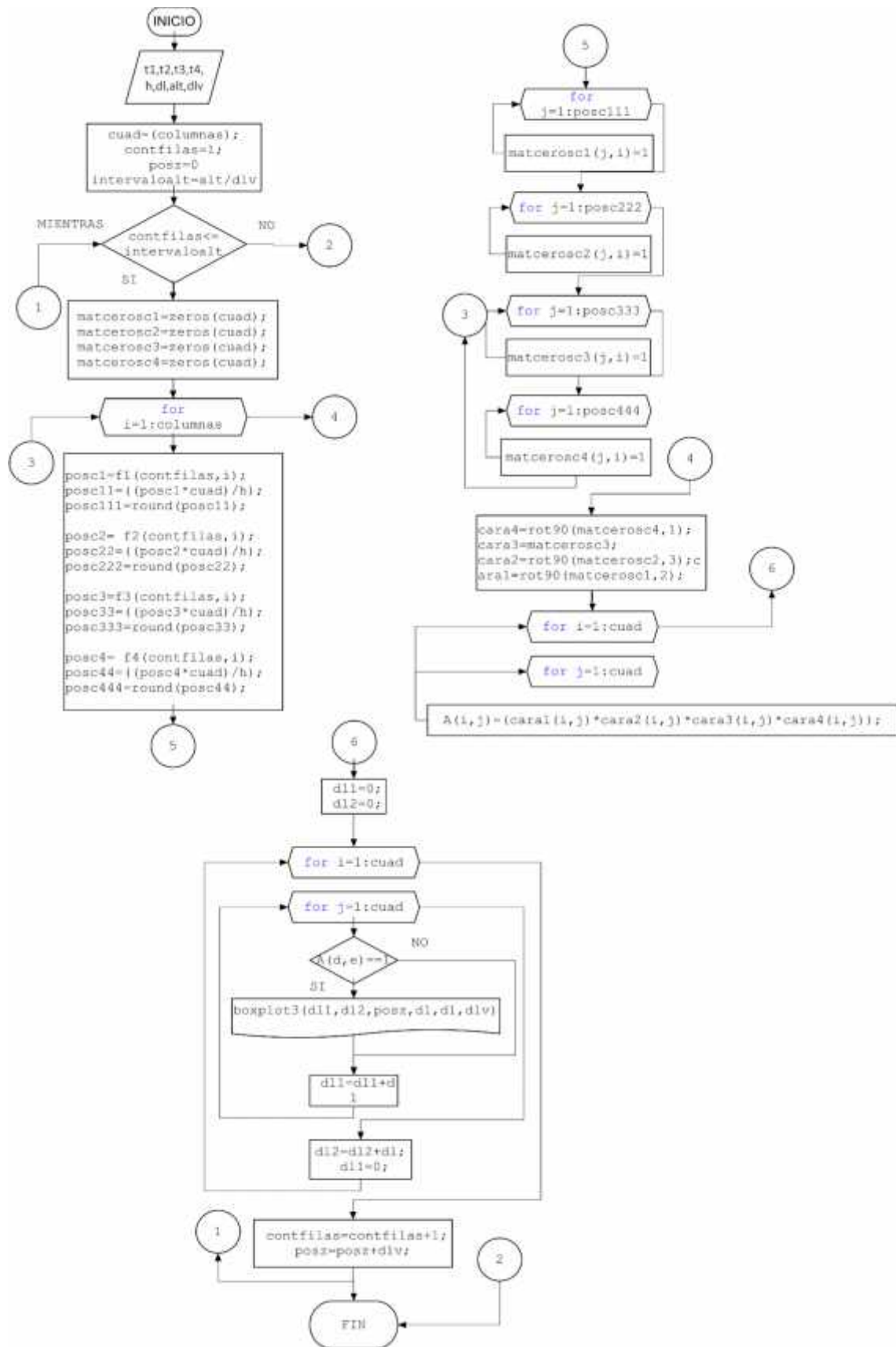
Donde, R es la matriz binaria y m el número de filas que contienen cada matriz de información almacenada (ver **Ecuación. 2-2**).



**Figura 12-2.** Reconstrucción del Objeto en 3D.

**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

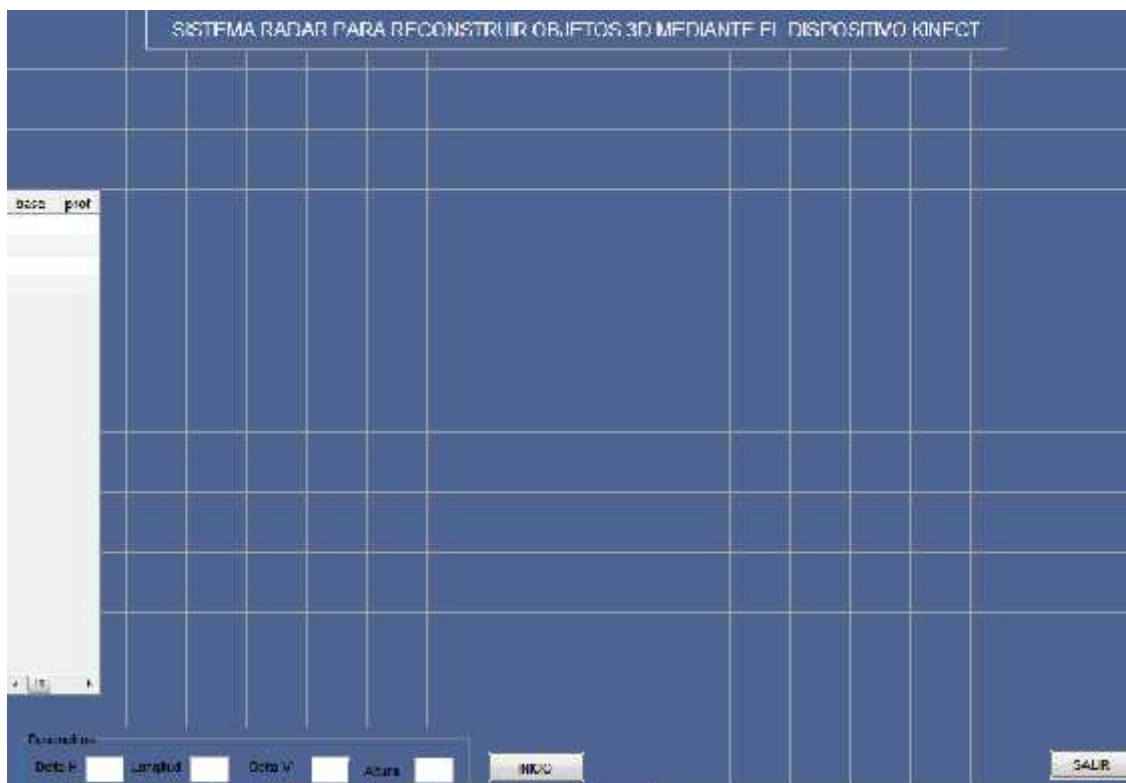
En la **Figura 13-2**, se observa el flujograma utilizado para reconstruir el objeto con datos de las matrices de profundidad de las cuatro caras laterales del cubo.



**Figura 13-2.** Flujograma para Reconstrucción del Objeto en 3D.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 2.3 Diseño de la Aplicación GUI

Se realizó la interfaz gráfica de usuario del sistema radar (ver **Figura 14-2**) el cual consta de componentes que permitan ingresar parámetros tales como: longitud de arista horizontal, Altura del Objeto y la distancia a la que se tomara cada medición horizontal y vertical para posteriormente y mediante el botón Inicio arranque el programa para realizar las mediciones hasta cubrir toda el área de reconstrucción, y finalmente visualizar el objeto en tres dimensiones.



**Figura 14-2.** Componentes de la Interfaz Gráfica de Usuario.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Luego de añadir los componentes al GUI de Matlab se procede a programar los callbacks mediante código Matlab. Esto se realiza dando click derecho sobre el componente de interés y seleccionando la opción Callback, como se observa en la **Figura 15-2**.

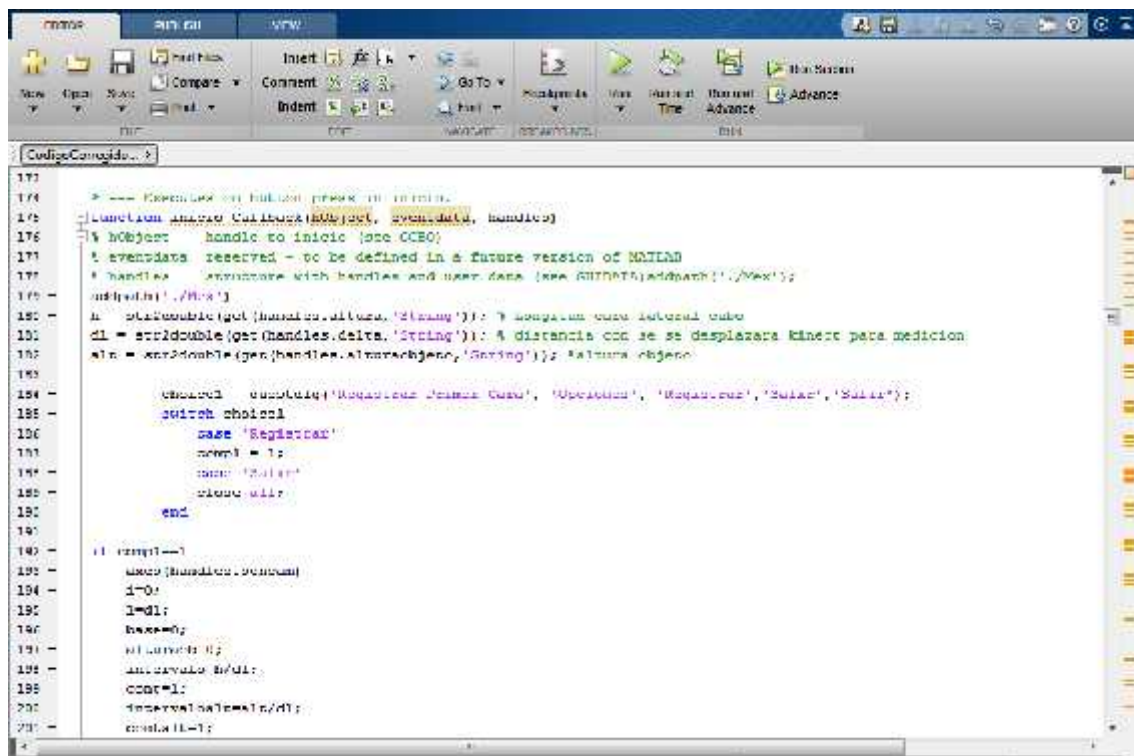
Una vez seleccionado la opción callback nos re direcciona al script (ver **Figura 16-2**) en donde se codifica los algoritmos planificados.

Finalizamos con la ejecución del GUI diseñado para la Reconstrucción del Objetos en 3D como muestra la **Figura 17-2** además en el **Anexo B** se muestra el Código Matlab del Programa.



**Figura 15-2.** Callback.

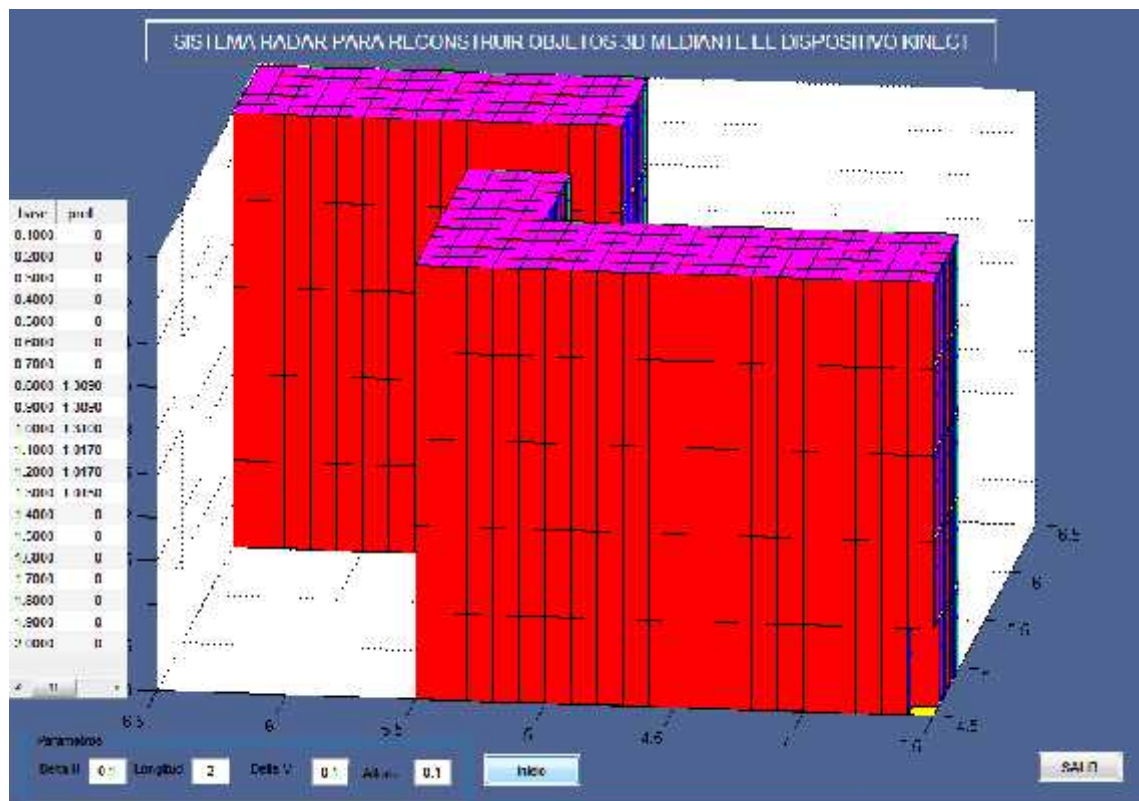
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 16-2.** Ventana Script para Codificación de Componentes.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016





**Figura 17-2.** GUI Sistema Radar para Reconstrucción de Objetos en 3D.

**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

## CAPITULO III

### 3. MEDICIONES Y RESULTADOS

#### 3.1 Objetos de Reconstrucción

Para realizar las pruebas de reconstrucción se tomó cuerpos de forma rectangular como muestra la **Figura 1-3** el cual permite facilitar las mediciones y el análisis de resultados realizado posteriormente.



**Figura 1–3.** Objeto en forma Rectangular.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

#### 3.2 Parámetros.

De acuerdo al análisis realizado en la tesis de grado “Sistema Radar Para Análisis Y Reconstrucción De Perfiles Topográficos A Media Escala” (Álvarez R., Shagñay S. 2015: p.84) cuyo análisis determinó que el sensor Kinect trabaja de mejor manera a una distancia de 2m, por



este motivo para las pruebas realizadas en los objetos se consideró que la longitud de aristas horizontales sea de 2m.

Se realizó pruebas de reconstrucción 3D en dos objetos diferentes para lo cual en la **Tabla 1-3** se describe los parámetros utilizados en cada uno de los objetos.

**Tabla 1-3.** Parámetros

PARAMETROS	OBJETO 1		OBJETO 2	
LONGITUD	2m	2m	2m	2m
DELTA HORIZONTAL	0.2m	0.1m	0.2m	0.1m
ALTURA	0.1m	0.1m	0.2m	0.2m
DELTA VERTICAL	0.1m	0.1m	0.1m	0.1m

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.2.1 Matrices de Datos Primer Objeto.

Para almacenar las matrices de las caras laterales con las medidas de profundidad del primer objeto (ver **Figura 2-3**) utilizamos los parámetros definidos en la **Tabla 1-3** para el objeto 1.



**Figura 2-3.** Objeto 1.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.2.1.1. Matrices de Datos Primer Objeto Prueba 1.

En la tabla la **Tabla 2-3**. Se muestra los datos medidos en cada cara del primer objeto para delta horizontal de 0.2m, delta altura 0.1m, altura de 0.1m y longitud de 2m.

**Tabla 2-3.** Matriz de Datos medidos del Primer Objeto Prueba 1.

DATOS DE PROFUNDIDAD					
Delta V.	Delta H.	Cara 1	Cara 2	Cara 3	Cara 4
0,1	0,2	0	0	0	0
0,1	0,4	0	0	0	0
0,1	0,6	0	0	0	0
0,1	0,8	0,99	1,186	0	1,309
0,1	1	0,989	1,1968	1,173	1,31
0,1	1,2	1,337	1,181	1,171	1,023
0,1	1,4	0	0,884	1,325	1,031
0,1	1,6	0	0	0	0
0,1	1,8	0	0	0	0
0,1	2	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

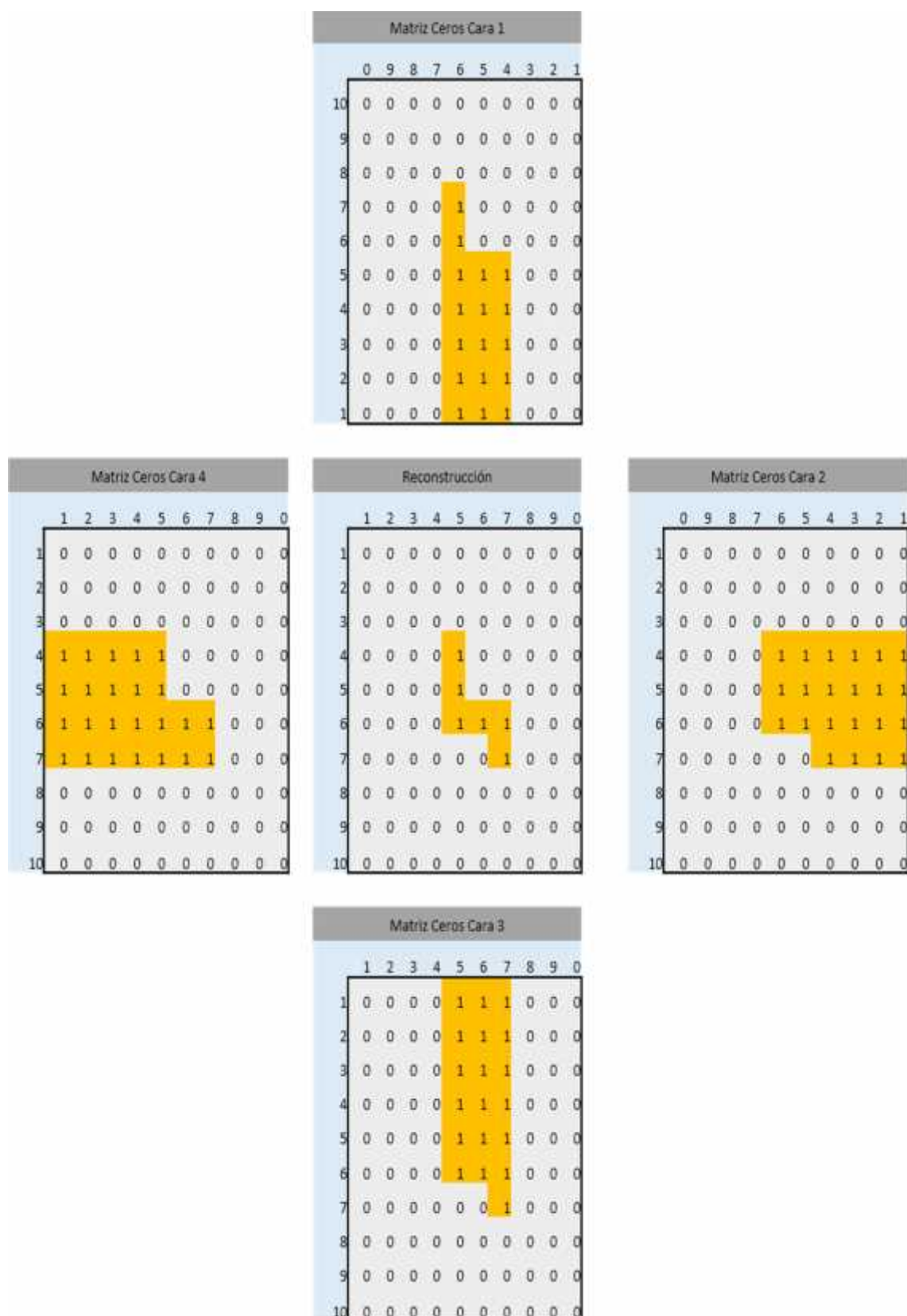
En la **Tabla 3-3**. Muestra los valores de las matrices de datos almacenadas de la prueba 1, el valor de la **Ecuación. 4-2** que representa la cantidad de 1's en matrices 0's y el redondeo del resultado obtenido.

**Tabla 3-3.** Cantidad de 1's Prueba 1.

NUMERO DE 1'S PARA REPRESENTAR MATRIZ DE RECONSTRUCCIÓN												
Dato	Matriz Cara 1			Matriz Cara 2			Matriz Cara 3			Matriz Cara 4		
	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0,99	4,95	5	1,186	5,93	6	0	0	0	1,309	6,545	7
5	0,989	4,945	5	1,1968	5,984	6	1,173	5,865	6	1,31	6,55	7
6	1,337	6,685	7	1,181	5,905	6	1,171	5,855	6	1,023	5,115	5
7	0	0	0	0,884	4,42	4	1,325	6,625	7	1,031	5,155	5
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 3-3**; Se muestra la asignación de 1's en las matrices de 0's para la prueba 1 del objeto 1, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8-2**.



**Figura 3-3.** Matrices de 0's con Asignación de 1's Primer Objeto Prueba 1.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.2.1.2 Matrices de Datos Primer Objeto Prueba 2.

En la tabla la **Tabla 4-3**. Se muestra los datos medidos en cada cara del objeto para delta horizontal de 0.1m, delta altura 0.1m, altura de 0.1m y longitud de 2m.

**Tabla 4-3:** Matriz de Datos medidos del Primer Objeto Prueba 2

DATOS DE PROFUNDIDAD					
Delta V.	Delta H.	Cara 1	Cara 2	Cara 3	Cara 4
0,1	0,1	0	0	0	0
0,1	0,2	0	0	0	0
0,1	0,3	0	0	0	0
0,1	0,4	0	0	0	0
0,1	0,5	0	0	0	0
0,1	0,6	0	0	0	0
0,1	0,7	0,993	1,181	0	1,309
0,1	0,8	0,994	1,181	0	1,309
0,1	0,9	0,993	1,179	1,182	1,31
0,1	1	0,994	1,18	1,188	1,301
0,1	1,1	1,313	1,181	1,181	1,017
0,1	1,2	1,309	1,18	1,181	1,017
0,1	1,3	0	0,884	1,317	1,0115
0,1	1,4	0	0,88	1,318	1,0115
0,1	1,5	0	0	0	0
0,1	1,6	0	0	0	0
0,1	1,7	0	0	0	0
0,1	1,8	0	0	0	0
0,1	1,9	0	0	0	0
0,1	2	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

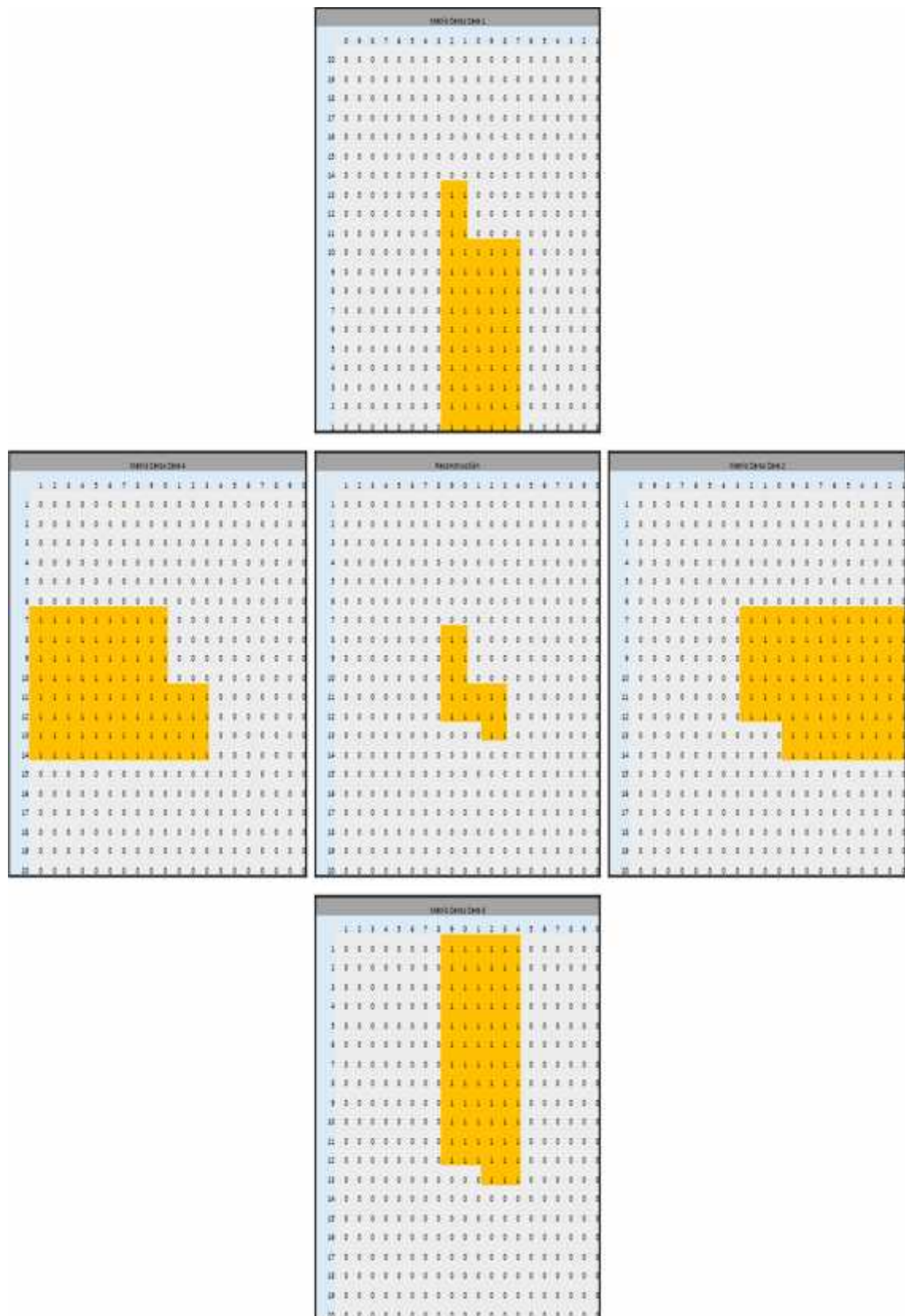
En la **Tabla 5-3**. Muestra los valores de las matrices de datos almacenadas de la prueba 2, el valor de **Ecuación. 4-2** que representa la cantidad de 1's en matrices 0's y el redondeo del resultado obtenido.

**Tabla 5-3.** Cantidad de 1's Prueba 2.

NUMERO DE 1'S PARA REPRESENTAR MATRIZ DE RECONSTRUCCIÓN												
Dato	Matriz Cara 1			Matriz Cara 2			Matriz Cara 3			Matriz Cara 4		
	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's	Medida	Regla 3	Cant 1's
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0,993	9,93	10	1,181	11,81	12	0	0	0	1,309	13,09	13
8	0,994	9,94	10	1,181	11,81	12	0	0	0	1,309	13,09	13
9	0,993	9,93	10	1,179	11,79	12	1,182	11,82	12	1,31	13,1	13
10	0,994	9,94	10	1,18	11,8	12	1,188	11,88	12	1,301	13,01	13
11	1,313	13,13	13	1,181	11,81	12	1,181	11,81	12	1,017	10,17	10
12	1,309	13,09	13	1,18	11,8	12	1,31	11,81	12	1,017	10,17	10
13	0	0	0	0,884	8,84	9	1,317	13,17	13	1,0115	10,115	10
14	0	0	0	0,88	8,8	9	1,318	13,18	13	1,0115	10,115	10
15	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 4-3**; Se muestra la asignación de 1's en las matrices de 0's para la prueba 2 del objeto 1, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8-2**



**Figura 4-3.** Asignación de 1's en la Matriz de 0's Primer Objeto Prueba 2.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



### 3.2.2 Matrices de Datos Segundo Objeto.

Para almacenar las matrices de información con las medidas del segundo objeto (ver **Figura 5-3**) utilizamos los parámetros definidos en la **Tabla 1-3**.



**Figura 5-3.** Objeto 2.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Se puede visualizar el Objeto 2 con mejor perspectiva en la **Figura 6-3**.



**Figura 6-3.** Objeto 2 Cara Posterior.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



### 3.2.2.1. Matrices de Datos Segundo Objeto Prueba 1

En la **Tabla 6-3**. Se muestra los datos medidos en cada cara del tercer objeto para delta horizontal de 0.2m, delta altura 0.1m, altura de 0.2m y longitud de 2m.

**Tabla 6-3:** Matriz de Datos medidos del Segundo Objeto Prueba 1

DATOS DE PROFUNDIDAD					
Delta V.	Delta H.	Cara 1	Cara 2	Cara 3	Cara 4
FILA 1					
0,1	0,2	0	0	0	0
0,1	0,4	0	0	0	0
0,1	0,6	0	0	0	0
0,1	0,8	0,99	1,186	0	1,309
0,1	1	0,989	1,197	1,173	1,31
0,1	1,2	1,337	1,181	1,171	1,023
0,1	1,4	0	0,884	1,325	1,031
0,1	1,6	0	0	0	0
0,1	1,8	0	0	0	0
0,1	2	0	0	0	0
FILA 2					
0,2	0,2	0	0	0	0
0,2	0,4	0	0	0	0
0,2	0,6	0	0	0	0
0,2	0,8	1,138	1,208	0	0
0,2	1	1,138	0,912	0,85	1,2879
0,2	1,2	1,335	0,912	1,148	1,287
0,2	1,4	0	0	1,148	1,08
0,2	1,6	0	0	0	0
0,2	1,8	0	0	0	0
0,2	2	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

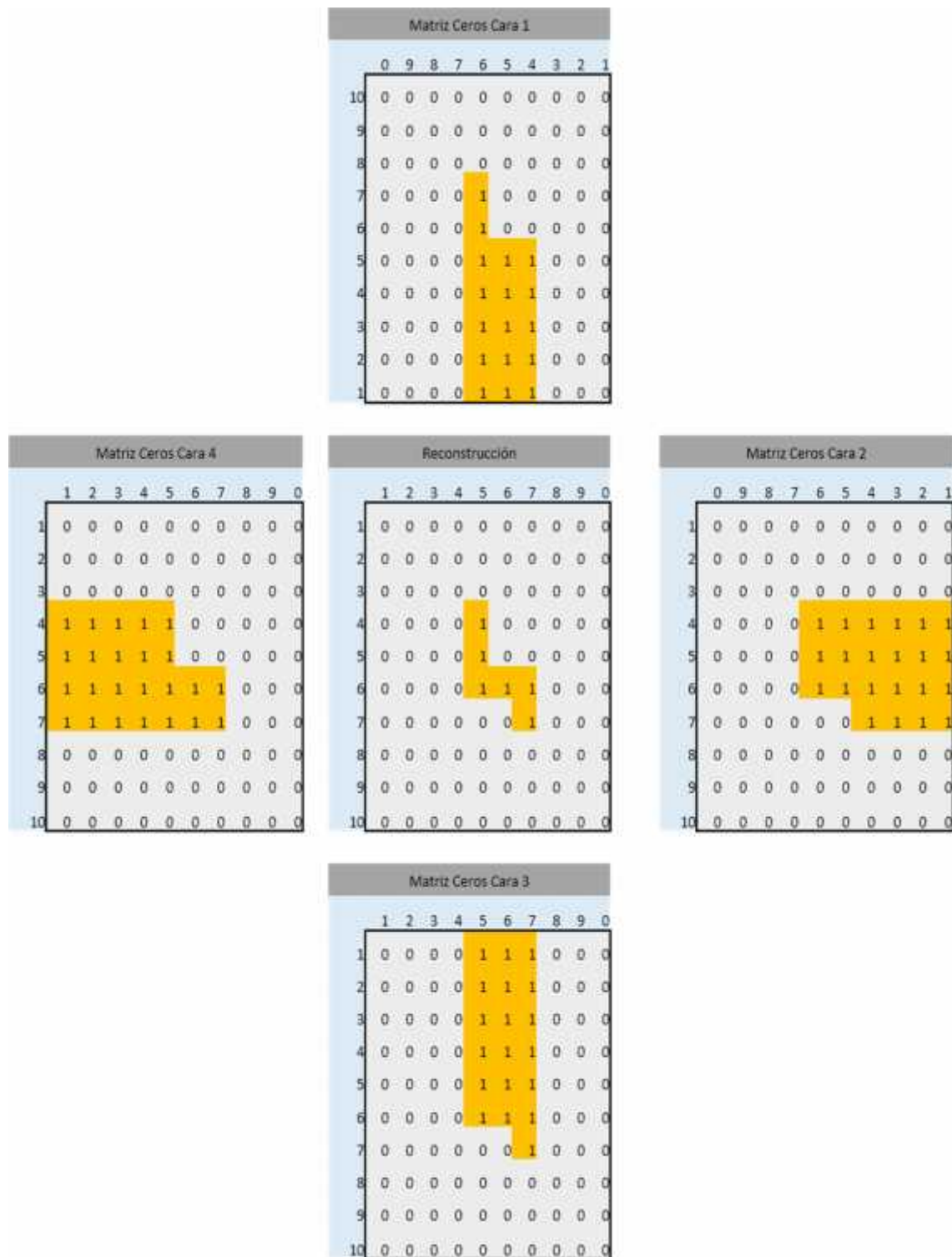
En la **Tabla 7-3**. Muestra los valores de las matrices de datos almacenadas de la prueba 1 del objeto 2, el valor de la **Ecuación. 4-2** que representa la cantidad de 1's en matrices 0's y el redondeo del resultado obtenido.

**Tabla 7-3.** Cantidad de 1's Prueba 1 Objeto 2.

Cornda	NUMERO DE 1'S PARA REPRESENTAR MATRIZ DE RECONSTRUCCIÓN											
	Matriz Cara 1			Matriz Cara 2			Matriz Cara 3			Matriz Cara 4		
	Medida	Regla 3	1's	Medida	Regla 3	1's	Medida	Regla 3	1's	Medida	Regla 3	1's
FILA 1												
(1.1)	0	0	0	0	0	0	0	0	0	0	0	0
(1.2)	0	0	0	0	0	0	0	0	0	0	0	0
(1.3)	0	0	0	0	0	0	0	0	0	0	0	0
(1.4)	0,99	4,95	5	1,186	5,93	6	0	0	0	1,309	6,545	7
(1.5)	0,989	4,945	5	1,1968	5,984	6	1,173	5,865	6	1,31	6,55	7
(1.6)	1,337	6,685	7	1,181	5,905	6	1,171	5,855	6	1,023	5,115	5
(1.7)	0	0	0	0,884	4,42	4	1,325	6,625	7	1,031	5,155	5
(1.8)	0	0	0	0	0	0	0	0	0	0	0	0
(1.9)	0	0	0	0	0	0	0	0	0	0	0	0
(1.10)	0	0	0	0	0	0	0	0	0	0	0	0
FILA 2												
(2.1)	0	0	0	0	0	0	0	0	0	0	0	0
(2.2)	0	0	0	0	0	0	0	0	0	0	0	0
(2.3)	0	0	0	0	0	0	0	0	0	0	0	0
(2.4)	1,138	5,69	6	1,208	6,04	6	0	0	0	0	0	0
(2.5)	1,138	5,69	6	0,912	4,56	5	0,85	4,25	4	1,2879	6,4395	6
(2.6)	1,335	6,675	7	0,912	4,56	5	1,148	5,74	6	1,287	6,435	6
(2.7)	0	0	0	0	0	0	1,148	5,74	6	1,08	5,4	5
(2.8)	0	0	0	0	0	0	0	0	0	0	0	0
(2.9)	0	0	0	0	0	0	0	0	0	0	0	0
(2.10)	0	0	0	0	0	0	0	0	0	0	0	0

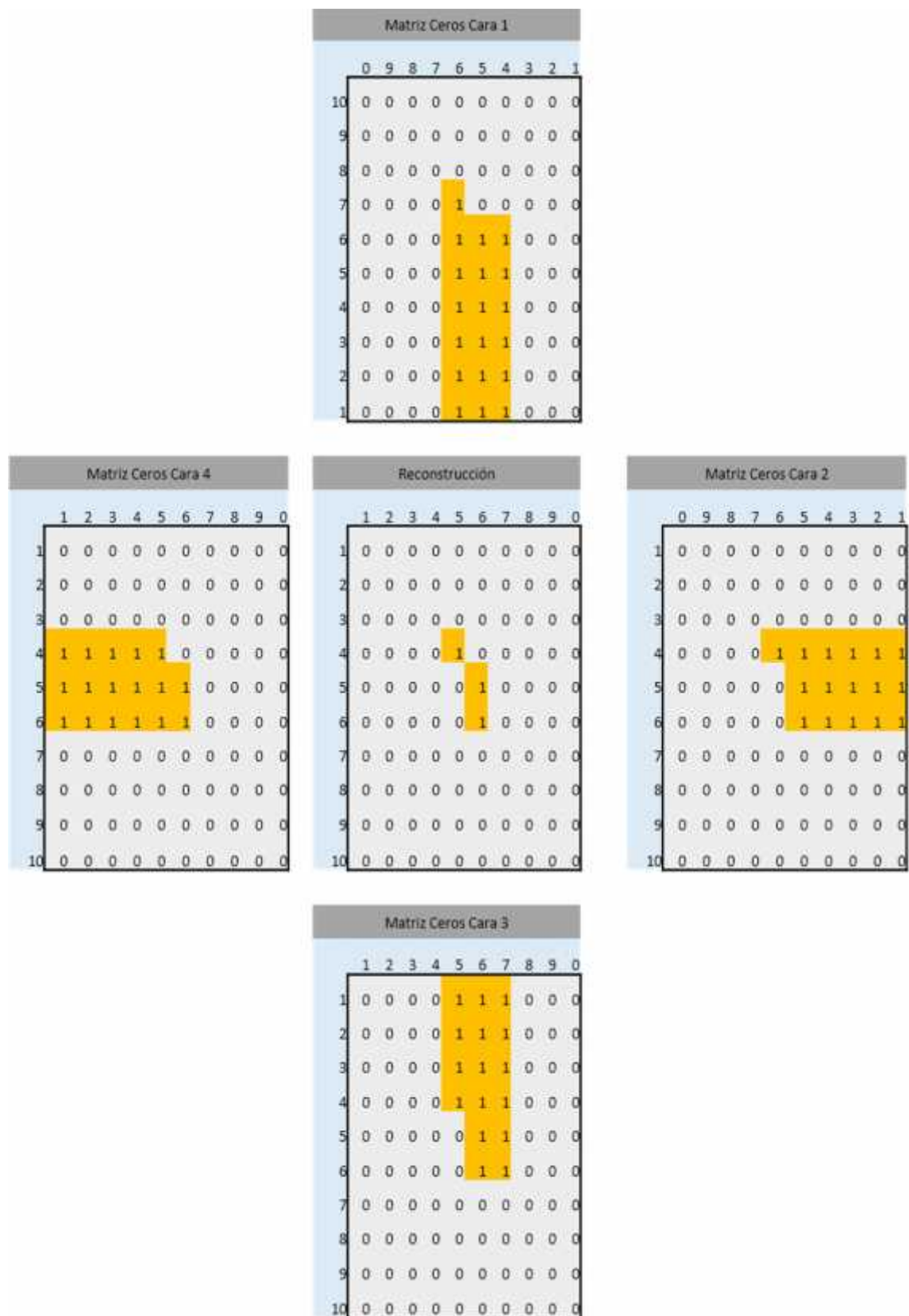
Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 7-3**; Se muestra la asignación de 1's en las matrices de 0's para la fila 1 de las matrices de datos obtenidas de la prueba 1 del objeto 2, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8-2**.



**Figura 7-3.** Matrices 0's Reasignado 1's Primera Filas de Matrices Almacenadas Prueba 1.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 8-3**; Se muestra la asignación de 1's en las matrices de 0's para la fila 2 de las matrices de datos obtenidas de la prueba 1 del objeto 2, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8** .



**Figura 8-3.** Matrices 0's Reasignado 1's Segunda Fila de Matrices Almacenadas Prueba 1.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.2.2.2. Matrices de Datos Segundo Objeto Prueba 2.

En la tabla la **Tabla 8-3**. Se muestra los datos medidos en cada cara del segundo objeto para delta horizontal de 0.1m, delta altura 0.1m, atura de 0.2m y longitud de 2m.

**Tabla 8-3:** Matriz de Datos medidos del Segundo Objeto Prueba 2.

DATOS DE PROFUNDIDAD					
Delta V.	Delta H.	Cara 1	Cara 2	Cara 3	Cara 4
<b>FILA 1</b>					
0,1	0,1	0	0	0	0
0,1	0,2	0	0	0	0
0,1	0,3	0	0	0	0
0,1	0,4	0	0	0	0
0,1	0,5	0	0	0	0
0,1	0,6	0	0	0	0
0,1	0,7	0,993	1,181	0	1,309
0,1	0,8	0,994	1,181	0	1,309
0,1	0,9	0,993	1,179	1,182	1,31
0,1	1	0,994	1,18	1,188	1,301
0,1	1,1	1,313	1,181	1,181	1,017
0,1	1,2	1,309	1,18	1,181	1,017
0,1	1,3	0	0,884	1,317	1,0115
0,1	1,4	0	0,88	1,318	1,0115
0,1	1,5	0	0	0	0
0,1	1,6	0	0	0	0
0,1	1,7	0	0	0	0
0,1	1,8	0	0	0	0
0,1	1,9	0	0	0	0
0,1	2	0	0	0	0
<b>FILA 2</b>					
0,2	0,1	0	0	0	0
0,2	0,2	0	0	0	0
0,2	0,3	0	0	0	0
0,2	0,4	0	0	0	0
0,2	0,5	0	0	0	0
0,2	0,6	0	0	0	0
0,2	0,7	0	1,208	0	0
0,2	0,8	1,138	1,208	0	0
0,2	0,9	1,138	1,21	0,85	1,2879
0,2	1	1,335	0,912	0,85	1,287
0,2	1,1	1,335	0,912	0,85	1,287
0,2	1,2	1,335	0,92	1,15	1,08
0,2	1,3	0	0	1,151	1,08
0,2	1,4	0	0	0	0
0,2	1,5	0	0	0	0
0,2	1,6	0	0	0	0
0,2	1,7	0	0	0	0
0,2	1,8	0	0	0	0
0,2	1,9	0	0	0	0
0,2	2	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Tabla 9-3**. Muestra los valores de las matrices de datos almacenadas de la prueba 2 del objeto 2, el valor de la **Ecuación. 4-2** que representa la cantidad de 1's en matrices 0's y el redondeo del resultado obtenido.

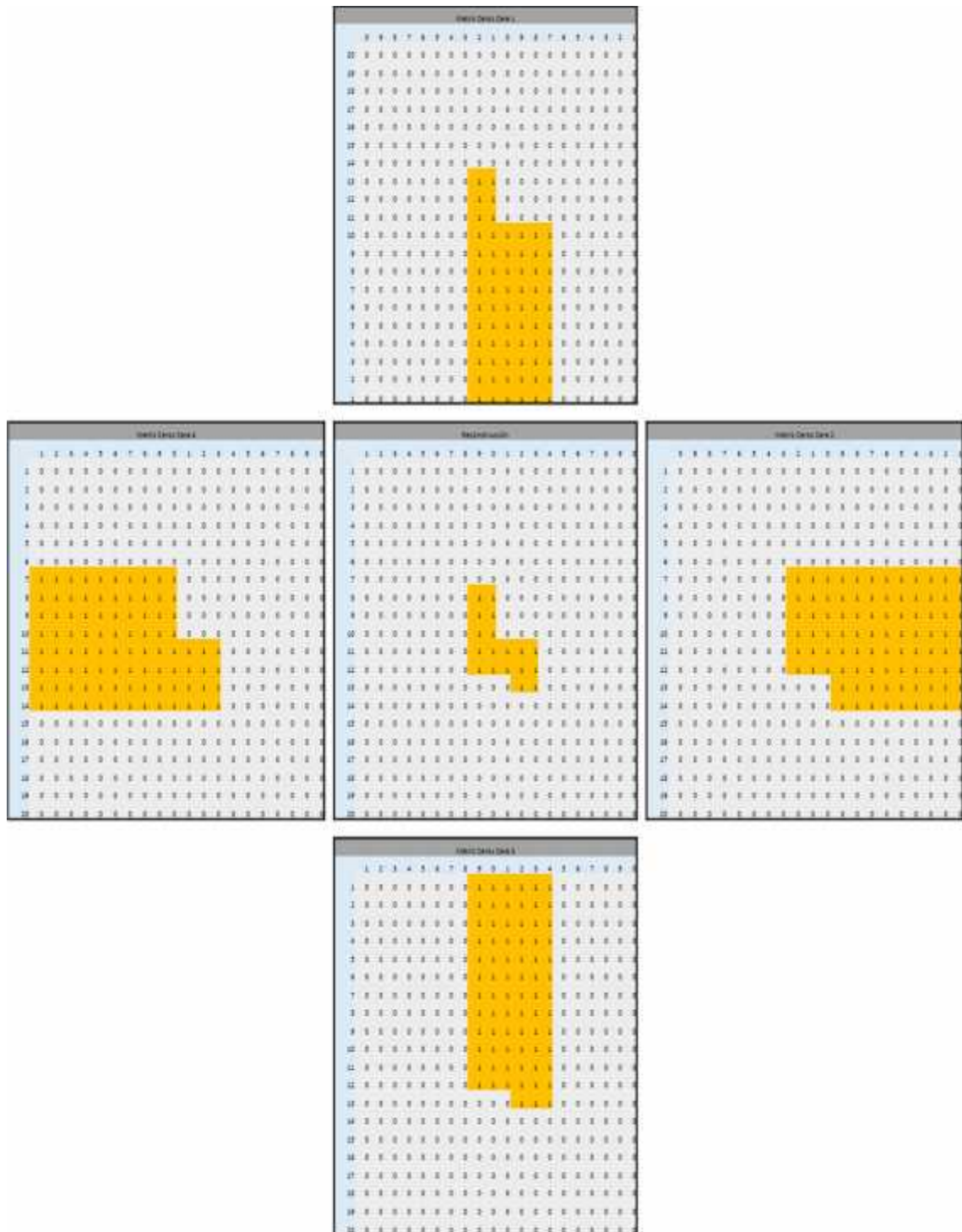
**Tabla 9-3.** Cantidad de 1's Prueba 2 Objeto 2.

Cornda	NUMERO DE 1'S PARA REPRESENTAR MATRIZ DE RECONSTRUCCIÓN											
	Matriz Cara 1			Matriz Cara 2			Matriz Cara 3			Matriz Cara 4		
	Medida	Regla 3	1's	Medida	Regla 3	1's	Medida	Regla 3	1's	Medida	Regla 3	1's
FILA 1												
(1.1)	0	0	0	0	0	0	0	0	0	0	0	0
(1.2)	0	0	0	0	0	0	0	0	0	0	0	0
(1.3)	0	0	0	0	0	0	0	0	0	0	0	0
(1.4)	0	0	0	0	0	0	0	0	0	0	0	0
(1.5)	0	0	0	0	0	0	0	0	0	0	0	0
(1.6)	0	0	0	0	0	0	0	0	0	0	0	0
(1.7)	0,993	9,93	10	1,181	11,81	12	0	0	0	1,309	13,09	13
(1.8)	0,994	9,94	10	1,181	11,81	12	0	0	0	1,309	13,09	13
(1.9)	0,993	9,93	10	1,179	11,79	12	1,182	11,82	12	1,31	13,1	13
(1.10)	0,994	9,94	10	1,18	11,8	12	1,188	11,88	12	1,301	13,01	13
(1.11)	1,313	13,13	13	1,181	11,81	12	1,181	11,81	12	1,017	10,17	10
(1.12)	1,309	13,09	13	1,18	11,8	12	1,31	11,81	12	1,017	10,17	10
(1.13)	0	0	0	0,884	8,84	9	1,317	13,17	13	1,0115	10,115	10
(1.14)	0	0	0	0,88	8,8	9	1,318	13,18	13	1,0115	10,115	10
(1.15)	0	0	0	0	0	0	0	0	0	0	0	0
(1.16)	0	0	0	0	0	0	0	0	0	0	0	0
(1.17)	0	0	0	0	0	0	0	0	0	0	0	0
(1.18)	0	0	0	0	0	0	0	0	0	0	0	0
(1.19)	0	0	0	0	0	0	0	0	0	0	0	0
(1.20)	0	0	0	0	0	0	0	0	0	0	0	0

FILA 2												
(2.1)	0	0	0	0	0	0	0	0	0	0	0	0
(2.2)	0	0	0	0	0	0	0	0	0	0	0	0
(2.3)	0	0	0	0	0	0	0	0	0	0	0	0
(2.4)	0	0	0	0	0	0	0	0	0	0	0	0
(2.5)	0	0	0	0	0	0	0	0	0	0	0	0
(2.6)	0	0	0	0	0	0	0	0	0	0	0	0
(2.7)	0	0	0	1,208	12,08	12	0	0	0	0	0	0
(2.8)	1,138	11,38	11	1,208	12,08	12	0	0	0	0	0	0
(2.9)	1,138	11,38	11	1,21	12,1	12	0,85	8,5	9	1,2879	12,879	13
(2.10)	1,335	13,35	13	0,912	9,12	9	0,85	8,5	9	1,287	12,87	13
(2.11)	1,335	13,35	13	0,912	9,12	9	0,85	8,5	9	1,287	12,87	13
(2.12)	1,335	13,35	13	0,92	9,2	9	1,148	11,5	12	1,08	10,8	11
(2.13)	0	0	0	0	0	0	1,148	11,51	12	1,08	10,8	11
(2.14)	0	0	0	0	0	0	0	0	0	1,08	10,8	11
(2.15)	0	0	0	0	0	0	0	0	0	0	0	0
(2.16)	0	0	0	0	0	0	0	0	0	0	0	0
(2.17)	0	0	0	0	0	0	0	0	0	0	0	0
(2.18)	0	0	0	0	0	0	0	0	0	0	0	0
(2.19)	0	0	0	0	0	0	0	0	0	0	0	0
(2.20)	0	0	0	0	0	0	0	0	0	0	0	0

Realizado por: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

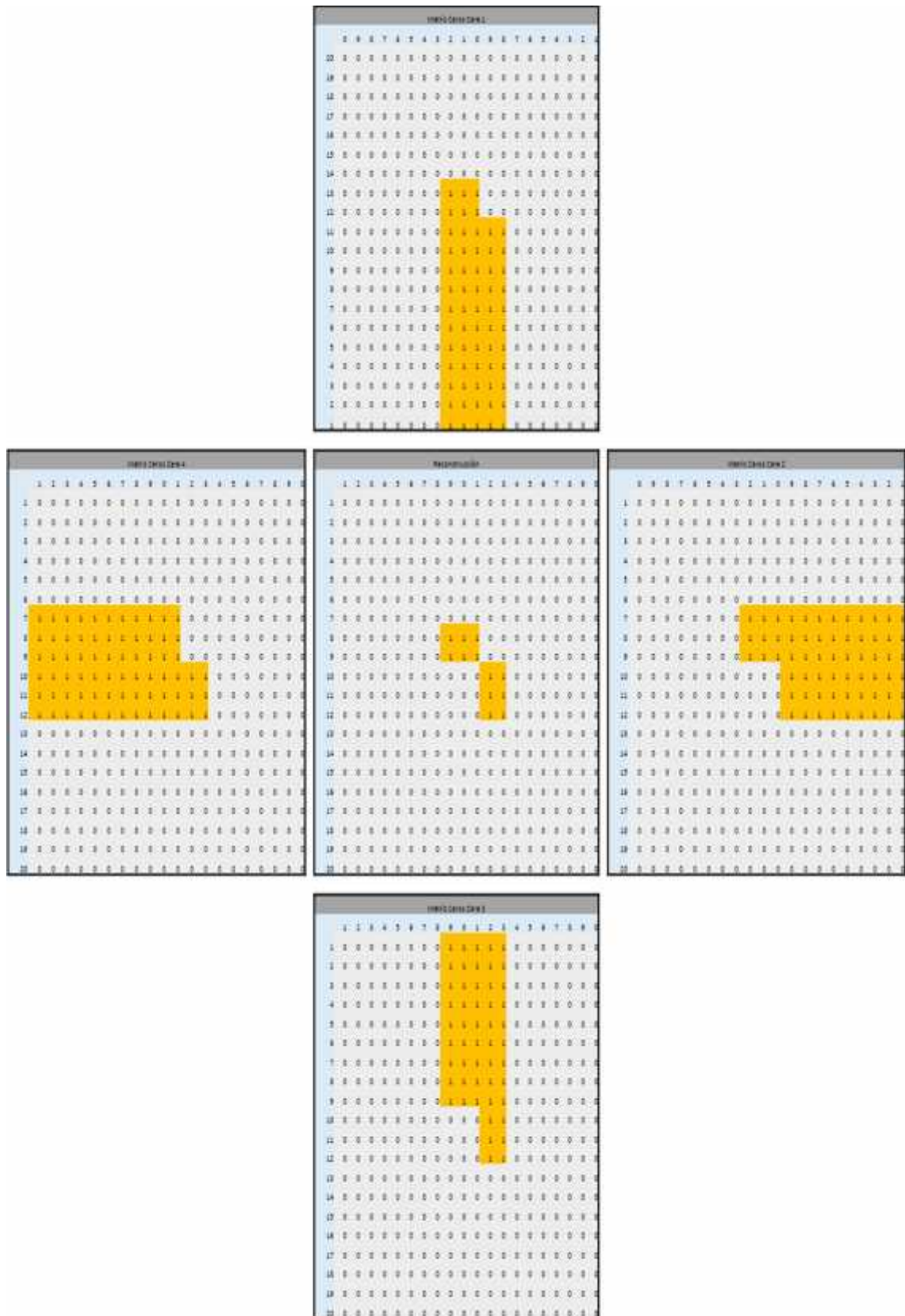
En la **Figura 9-3**; Se muestra la asignación de 1's en las matrices de 0's para la fila 1 de las matrices de datos obtenidas de la prueba 2 del objeto 2, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8-2**.



**Figura 9-3.** Matrices 0's Reasignado 1's Primera Fila de Matrices Almacenadas Prueba 2.  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

En la **Figura 10-3**; Se muestra la asignación de 1's en las matrices de 0's para la fila 2 de las matrices de datos obtenidas de la prueba 2 del objeto 2, las matrices se encuentran rotadas para entender la matriz de reconstrucción de objeto de acuerdo a la **Ecuación. 8** .





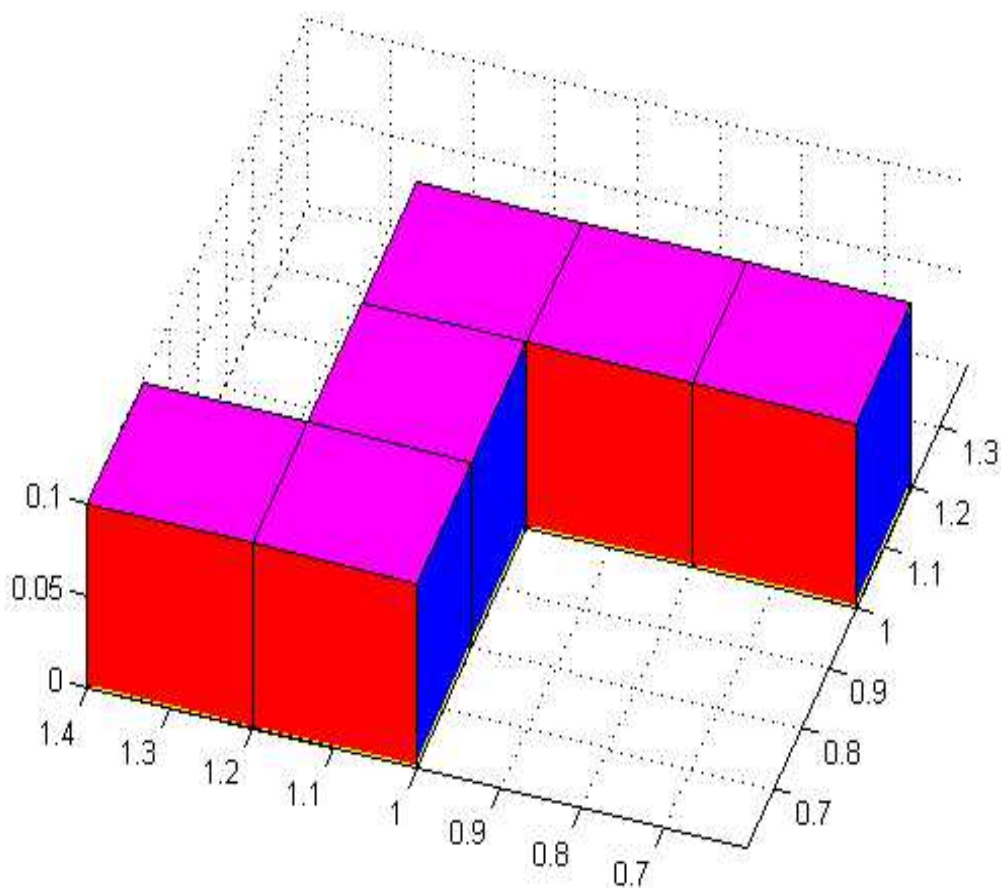
**Figura 10-3.** Matrices 0's Reasignado 1's Segunda Fila de Matrices Almacenadas Prueba 2.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.3 Reconstrucción de Objetos en 3D.

Para la gráfica de reconstrucción de objetos se define que cada dato con valor 1 representa una fracción del objeto en hexaedro o cubo, para el conjunto de datos que contiene la matriz binaria R.

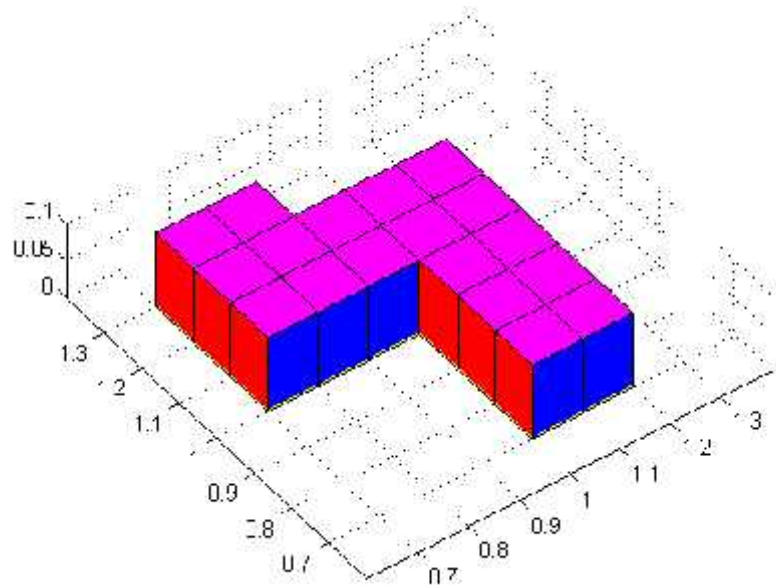
#### 3.3.1 Reconstrucción 3D Primer Objeto.

La reconstrucción tridimensional del objeto 1 (ver **Figura 2-3**) en la prueba 1 muestra la **Figura 11-3**



**Figura 11-3.** Reconstrucción Objeto 1 Prueba 1.  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

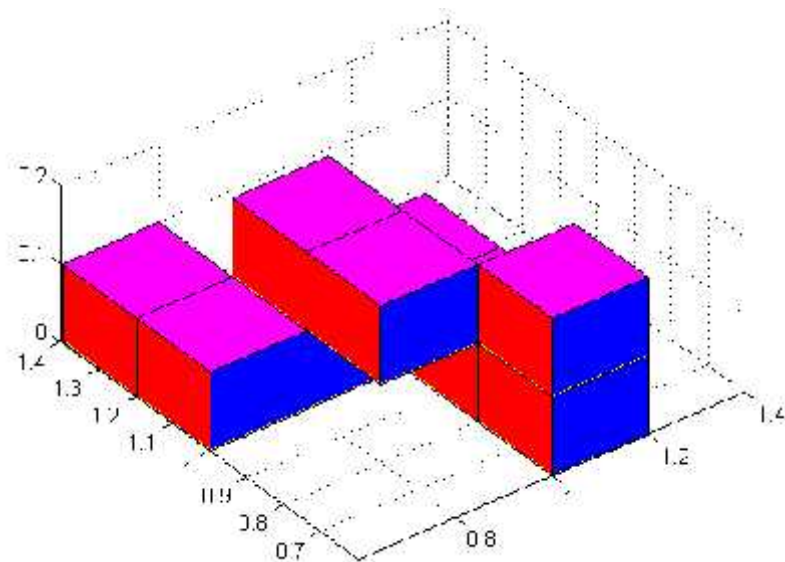
La reconstrucción tridimensional del objeto 1 (ver **Figura 2-3**) en la prueba 2 muestra la **Figura 12-3**.



**Figura 12-3.** Reconstrucción Objeto 1 Prueba 2.  
 Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

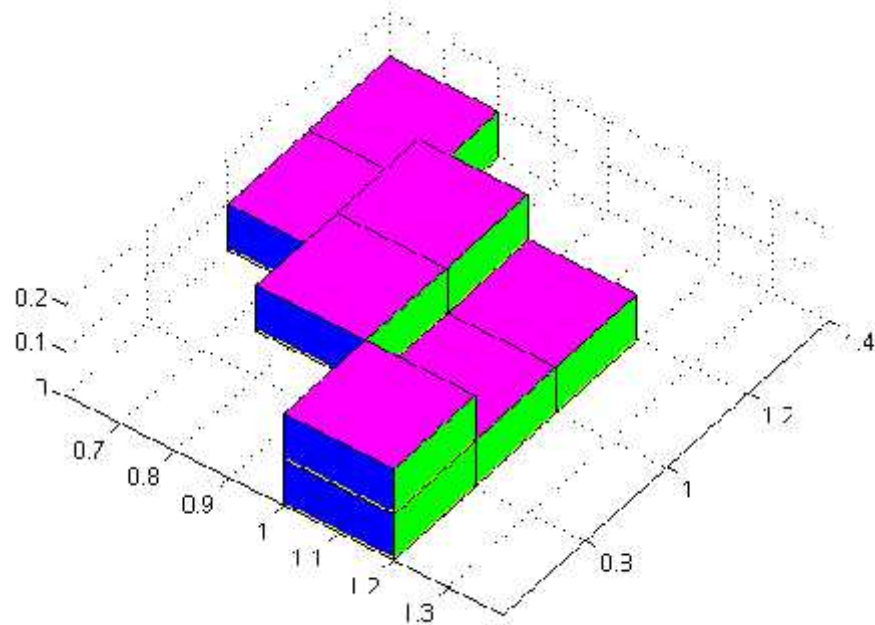
### 3.3.2 Reconstrucción 3D Segundo Objeto.

La reconstrucción tridimensional del objeto 2 (ver **Figura 5-3**) en su prueba 1 muestra la **Figura 13-3** y en la **Figura 14-3** se muestra la reconstrucción del objeto desde otra perspectiva para visualizar mejor el objeto 3 reconstruido.

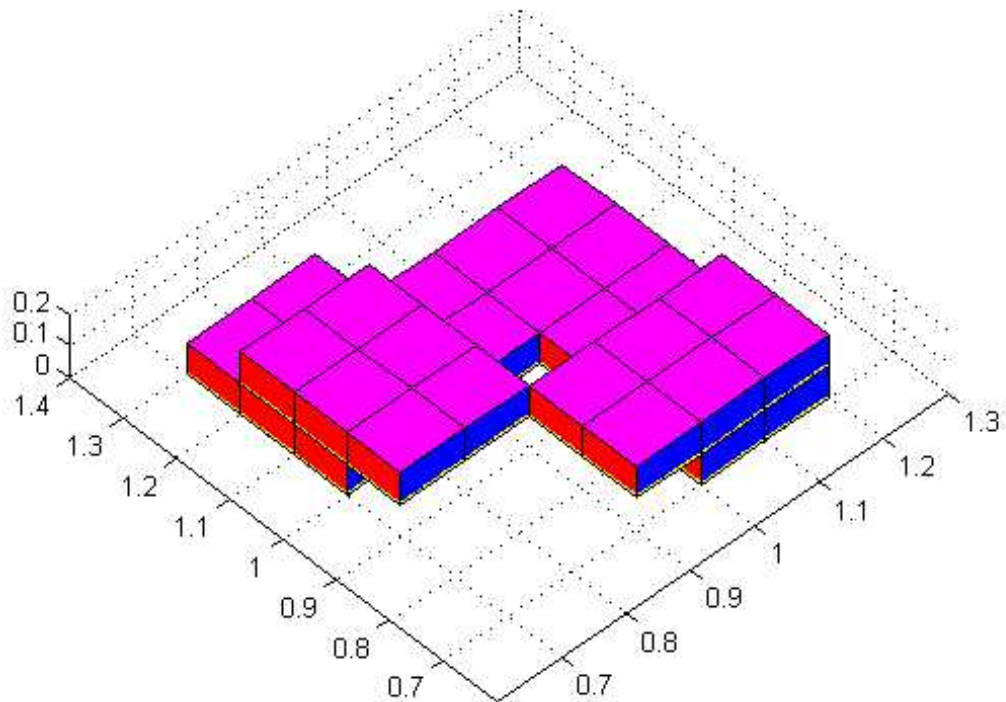


**Figura 13-3.** Reconstrucción Objeto 2 Prueba 1.  
 Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

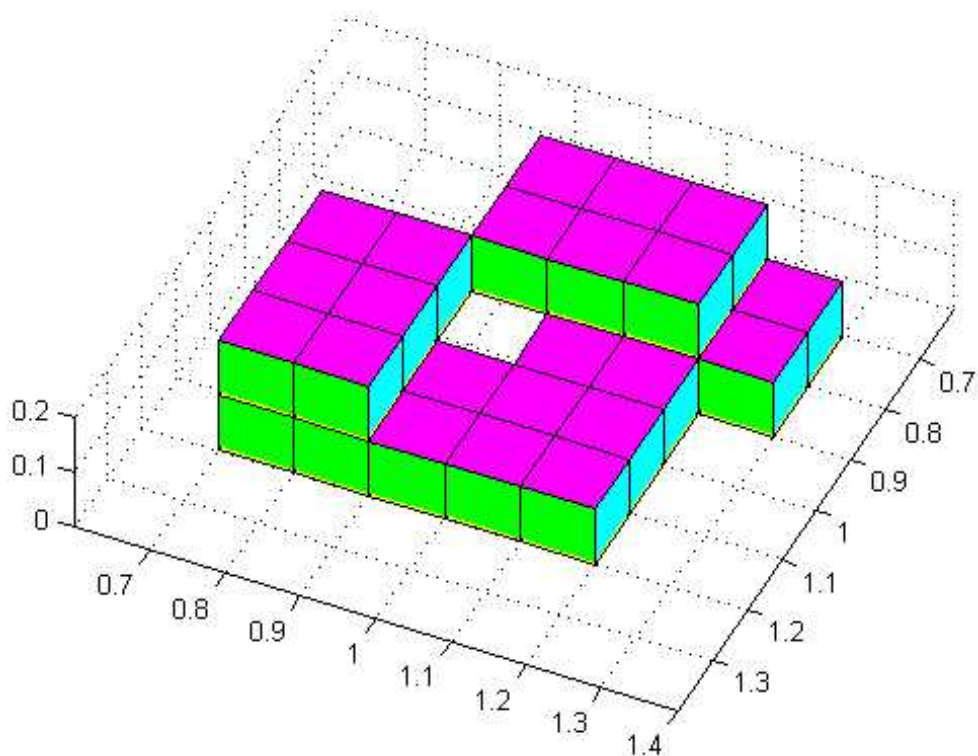
La reconstrucción tridimensional del objeto 2 (ver **Figura 5-3**) en su prueba 2 muestra la **Figura 15-3** y en la **Figura 16-3** se muestra la reconstrucción del objeto desde otra perspectiva para visualizar mejor el objeto 3 reconstruido.



**Figura 14-3.** Reconstrucción Objeto 2 Prueba 1 vista 2.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 15-3.** Reconstrucción Objeto 2 Prueba 2.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 16-3.** Reconstrucción Objeto 2 Prueba 2 vista 2.  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

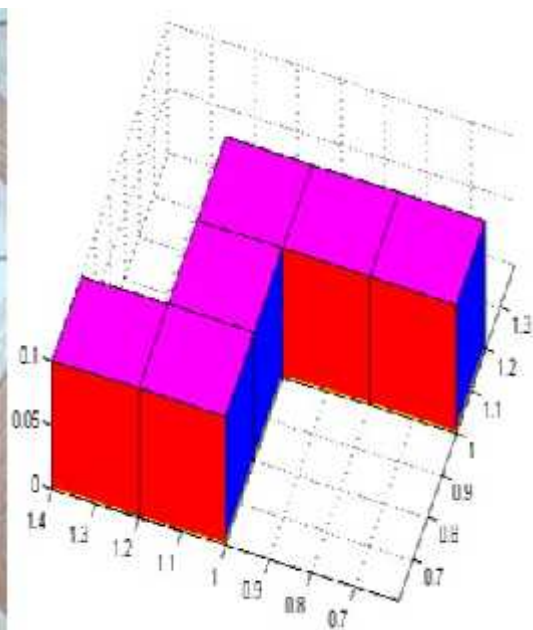
### 3.4 Análisis Comparativo.

#### 3.4.1 Análisis Objeto 1.

La **Figura 17-3** muestra el objeto 1 y la reconstrucción del objeto para mediciones a desplazamiento horizontal de 0.2m, desplazamiento vertical de 0.1m para altura de 0.1m en un área de reconstrucción de 2 m<sup>2</sup>.

No obstante se evidencia que cada caja de cartón que forma parte del objeto 1 está representado en la reconstrucción por 2 cubos de manera que existe una coincidencia de 2 datos por el ancho y 1 dato por el largo de cada caja además de evidenciar que existe una altura de 0.1m para todo el objeto determinando que las matrices que contienen la información de profundidad tan solo tiene una fila de datos.

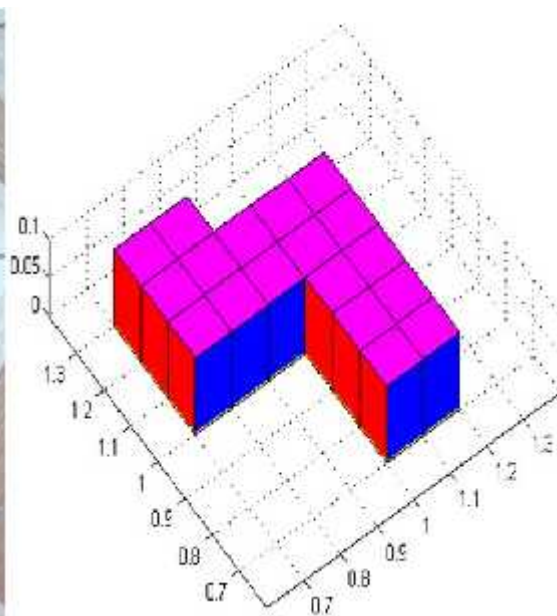




**Figura 17-3.** Objeto 1 Real y Reconstrucción 3D Prueba 1.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

La **Figura 18-3** muestra el objeto 1 y la reconstrucción del objeto para mediciones a desplazamiento horizontal de 0.1m, desplazamiento vertical de 0.1m para altura de 0.1m en un área de reconstrucción de 2 m<sup>2</sup>.



**Figura 18-3.** Objeto 1 Real y Reconstrucción 3D Prueba 2.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Sin embargo se evidencia que cada caja de cartón que es parte del objeto 1 está representado en la reconstrucción por 6 cubos de manera que existe una coincidencia de 3 datos por el ancho y 2 datos por el largo de cada caja además de evidenciar que existe una altura de 0.1m para el objeto constatando que las matrices que contienen las mediciones de profundidad tan solo tiene una fila de datos.

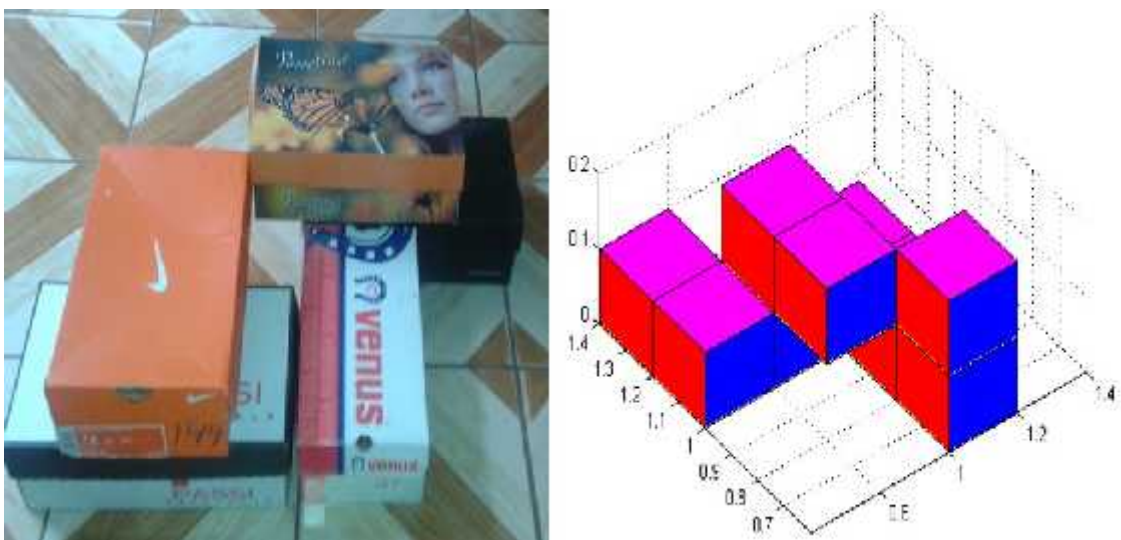
### 3.4.3. Análisis Objeto 2.

La **Figura 19-3** muestra el objeto 2 y su reconstrucción para desplazamiento horizontal de 0.2m, desplazamiento vertical de 0.1m, altura de 0.2m en un área de reconstrucción de 2 m<sup>2</sup>.

Ahora bien se observa que la reconstrucción del objeto no se aproxima al real debido a que los cubos que reconstruyen no son lo suficiente para reconstruir el objeto pero se observa que existe una altura de 0.2m para todo el objeto de manera que las matrices que contienen la información de profundidad tiene dos filas de datos.

La **Figura 20-3** muestra el objeto 2 y su reconstrucción para desplazamiento horizontal de 0.1m, desplazamiento vertical de 0.1m, altura de 0.2m en un área de reconstrucción de 2 m<sup>2</sup>.

Para este caso se observa que la reconstrucción del objeto se aproxima al real debido a que los cubos que reconstruyen al objeto son lo suficiente para reconstruirlo asimismo se observa que existe una altura de 0.2m para todo el objeto de manera que las matrices que contienen la información de profundidad tiene dos filas de datos.



**Figura 19-3.** Objeto 2 Real y Reconstrucción 3D Prueba 1.

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 20-3.** Objeto 2 Real y Reconstrucción 3D Prueba 2.

**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Ahora bien se constata al final del análisis comparativo que para desplazamientos horizontales de 0.2m en los tres objetos cada caja de cartón obtiene una longitud de 0.4m×0.2m siendo las longitudes aproximadamente reales a las cajas de 0.33m×0.185m por otra parte para desplazamientos horizontales de 0.1m los objetos se reconstruyen con mayor aproximación al objeto real debido a que las longitudes de cada caja ahora son de 0.3m×0.2m.

### 3.5 Concatenación de Datos

Se ha diseñado un algoritmo para dimensionar cada dato de las matrices de información de las caras laterales del objeto, con la finalidad de obtener cada dato dimensionado a una sub-matriz de (5\*5) redimensionando las matrices de información a ((n\*5) × (m\*5)), esto nos permitirá que en objetos curvilíneos la visualización de la reconstrucción 3D sea similar al objeto real.

#### Matriz Normal

$$A = \begin{pmatrix} aa_{1,1} & ab_{1,2} & \dots & az_{1,n} \\ ba_{2,1} & bb_{2,2} & \dots & bz_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ za_{n,1} & zb_{n,2} & \dots & zz_{n,n} \end{pmatrix}$$



### Matriz Concatenada

A=

$$\begin{pmatrix} \begin{pmatrix} aa_{11} & \dots & aa_{51} \\ \vdots & \ddots & \vdots \\ aa_{51} & \dots & aa_{55} \end{pmatrix}_{1,1} & \begin{pmatrix} ab_{11} & \dots & ab_{51} \\ \vdots & \ddots & \vdots \\ ab_{51} & \dots & ab_{55} \end{pmatrix}_{1,2} & \dots & \begin{pmatrix} az_{11} & \dots & az_{51} \\ \vdots & \ddots & \vdots \\ az_{51} & \dots & az_{55} \end{pmatrix}_{1,n} \\ \begin{pmatrix} ba_{11} & \dots & ba_{51} \\ \vdots & \ddots & \vdots \\ ba_{51} & \dots & ba_{55} \end{pmatrix}_{2,1} & \begin{pmatrix} bb_{11} & \dots & bb_{51} \\ \vdots & \ddots & \vdots \\ bb_{51} & \dots & bb_{55} \end{pmatrix}_{2,2} & \dots & \begin{pmatrix} bz_{11} & \dots & bz_{51} \\ \vdots & \ddots & \vdots \\ bz_{51} & \dots & bz_{55} \end{pmatrix}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{pmatrix} za_{11} & \dots & za_{51} \\ \vdots & \ddots & \vdots \\ za_{51} & \dots & za_{55} \end{pmatrix}_{n,1} & \begin{pmatrix} zb_{11} & \dots & zb_{51} \\ \vdots & \ddots & \vdots \\ zb_{51} & \dots & zb_{55} \end{pmatrix}_{n,2} & \dots & \begin{pmatrix} zz_{11} & \dots & zz_{51} \\ \vdots & \ddots & \vdots \\ zz_{51} & \dots & zz_{55} \end{pmatrix}_{n,n} \end{pmatrix}_{((n*5) \times (n*5))}$$

Para evidenciar el funcionamiento de la reconstrucción de objetos en forma tridimensional con datos concatenados de las mediciones realizadas en las caras laterales del objeto, se reconstruyo los mismos objetos anteriormente probados exponiendo los resultados a continuación.

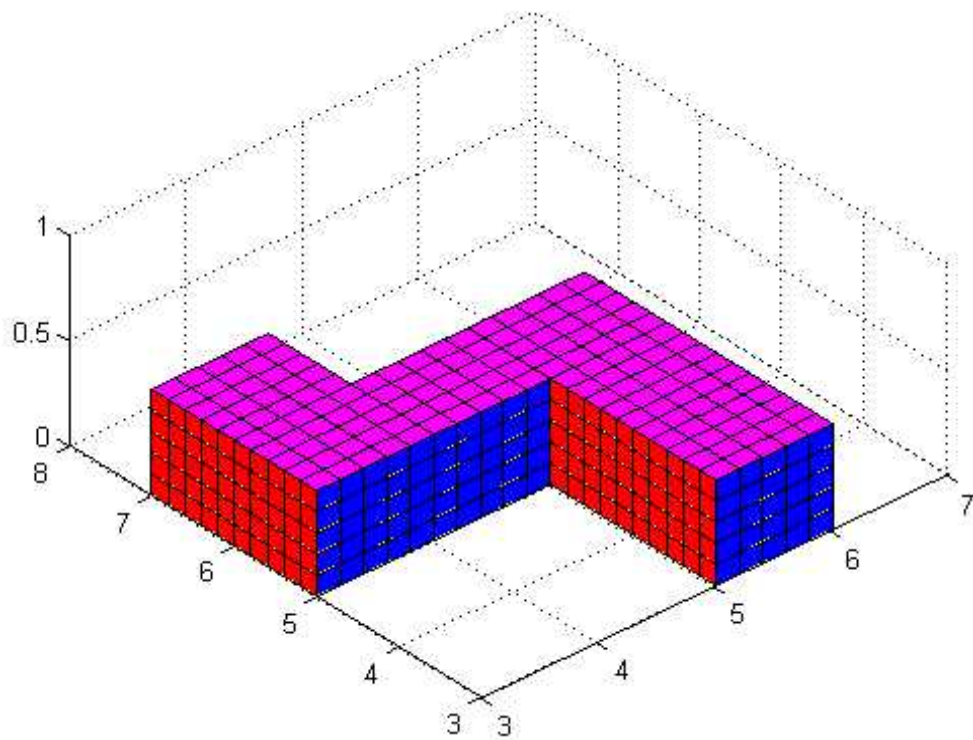
#### 3.5.1 Reconstrucción Objeto 1 con Concatenación.

En la **Figura 21-3** se exhibe la reconstrucción del objeto 1 (ver **Figura 2-3**) con datos concatenados de la **Tabla 3-3** correspondiente a la prueba 1 del objeto y en la **Figura 22-3** se exhibe la reconstrucción del objeto 1 con datos concatenados de la **Tabla 5-3** correspondiente a la prueba 2 del objeto.

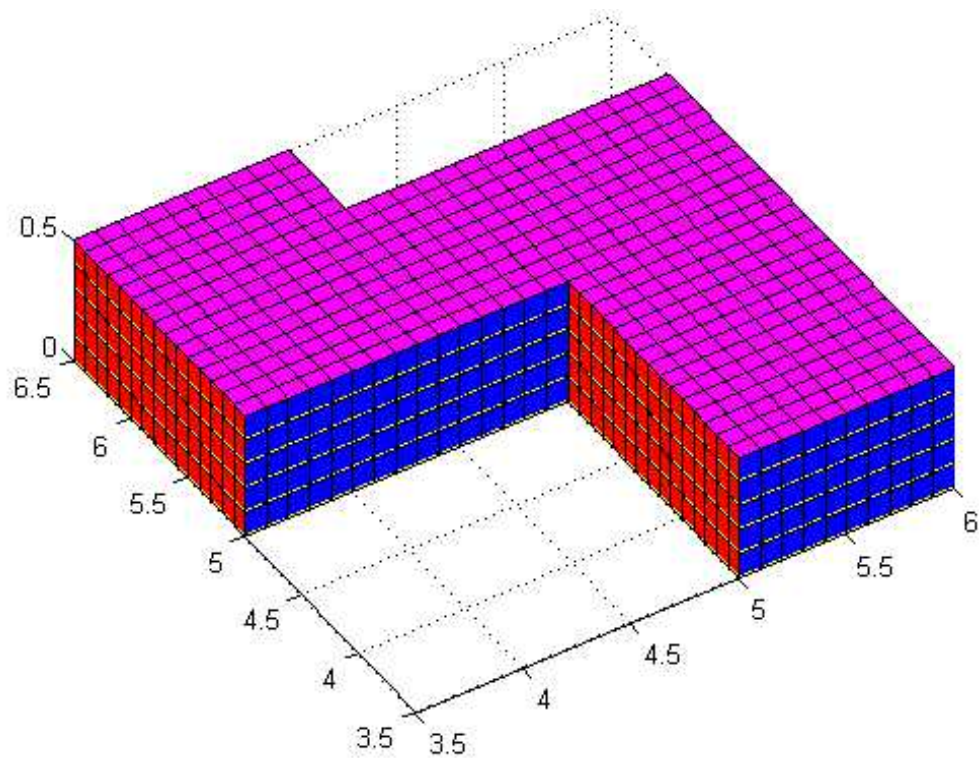
#### 3.5.2 Reconstrucción de Objeto 2 con Concatenación de Datos.

En la **Figura 23-3** se exhibe la reconstrucción del objeto 2 (ver **Figura 5-3**) desde la vista frontal y vista posterior con datos concatenados de la **Tabla 7-3** correspondiente a la prueba 1 del objeto.

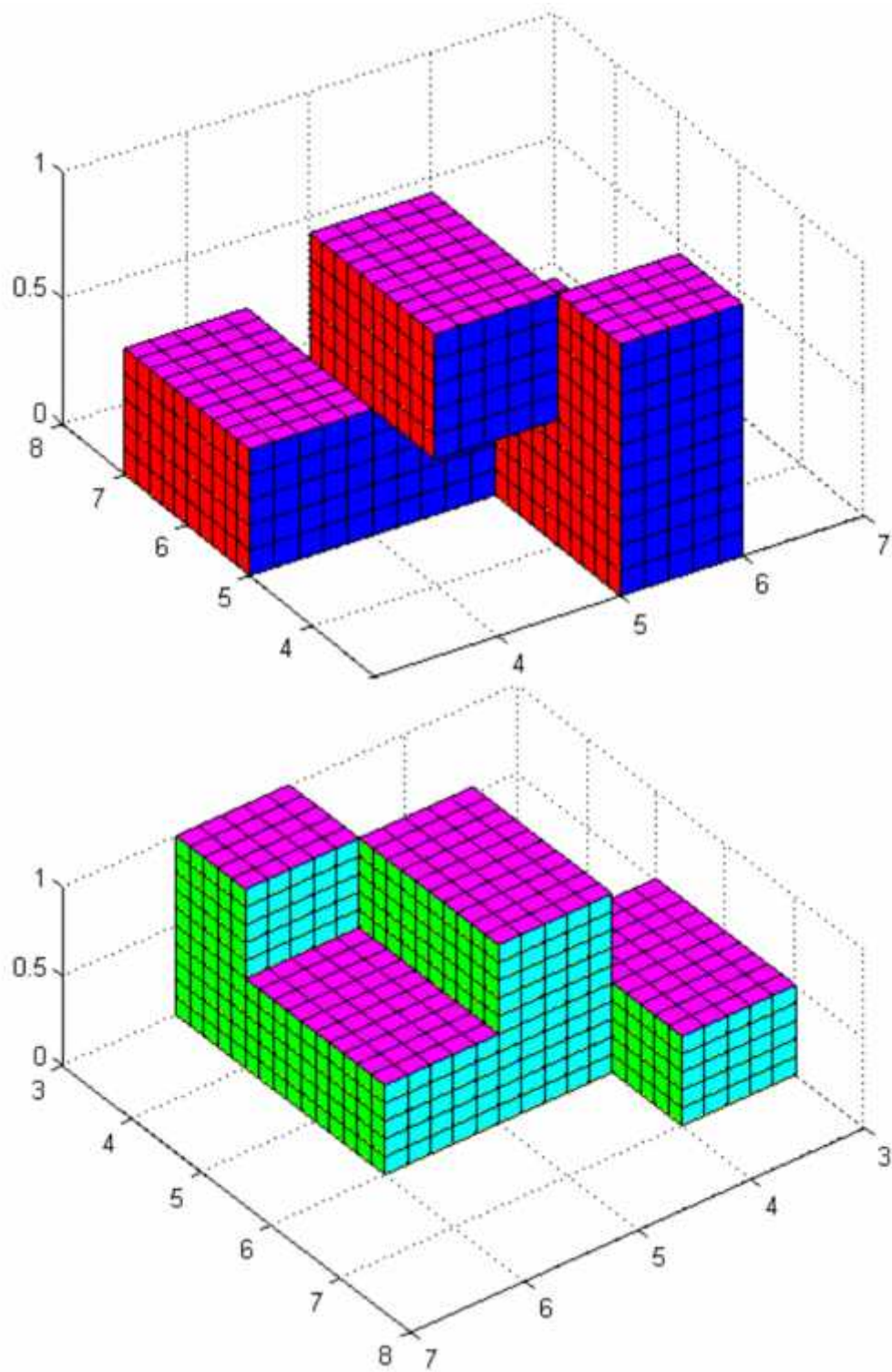
En la **Figura 24-3** se exhibe la reconstrucción del objeto 2 (ver **Figura 5-3**) desde la vista frontal y vista posterior con datos concatenados de la **Tabla 9-3** correspondiente a la prueba 2 del objeto.



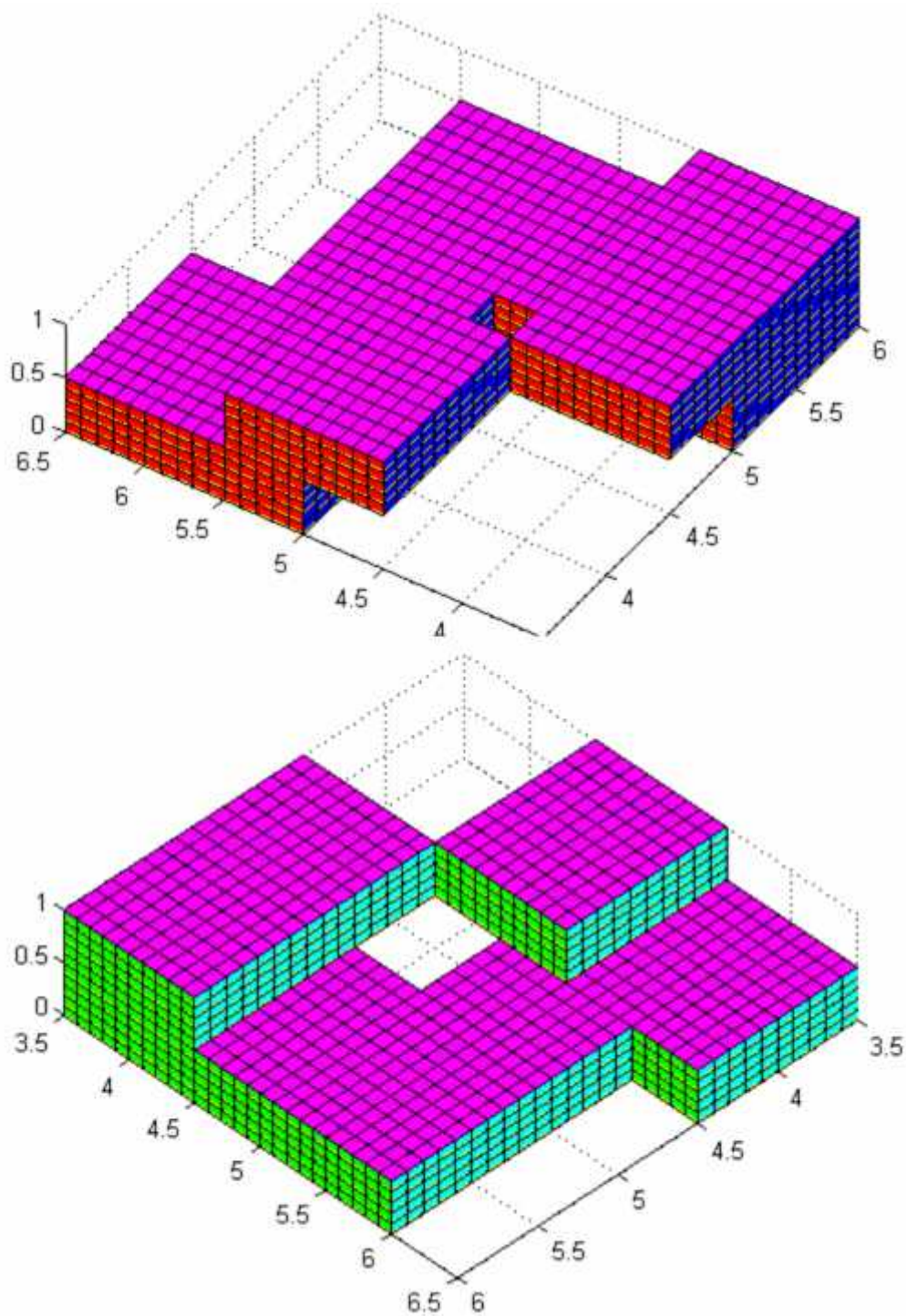
**Figura 21-3.** Reconstrucción 3D Objeto 1 con datos concatenados **Tabla 3-3.**  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 22-3.** Reconstrucción 3D Objeto 1 con datos concatenados **Tabla 5-3.**  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



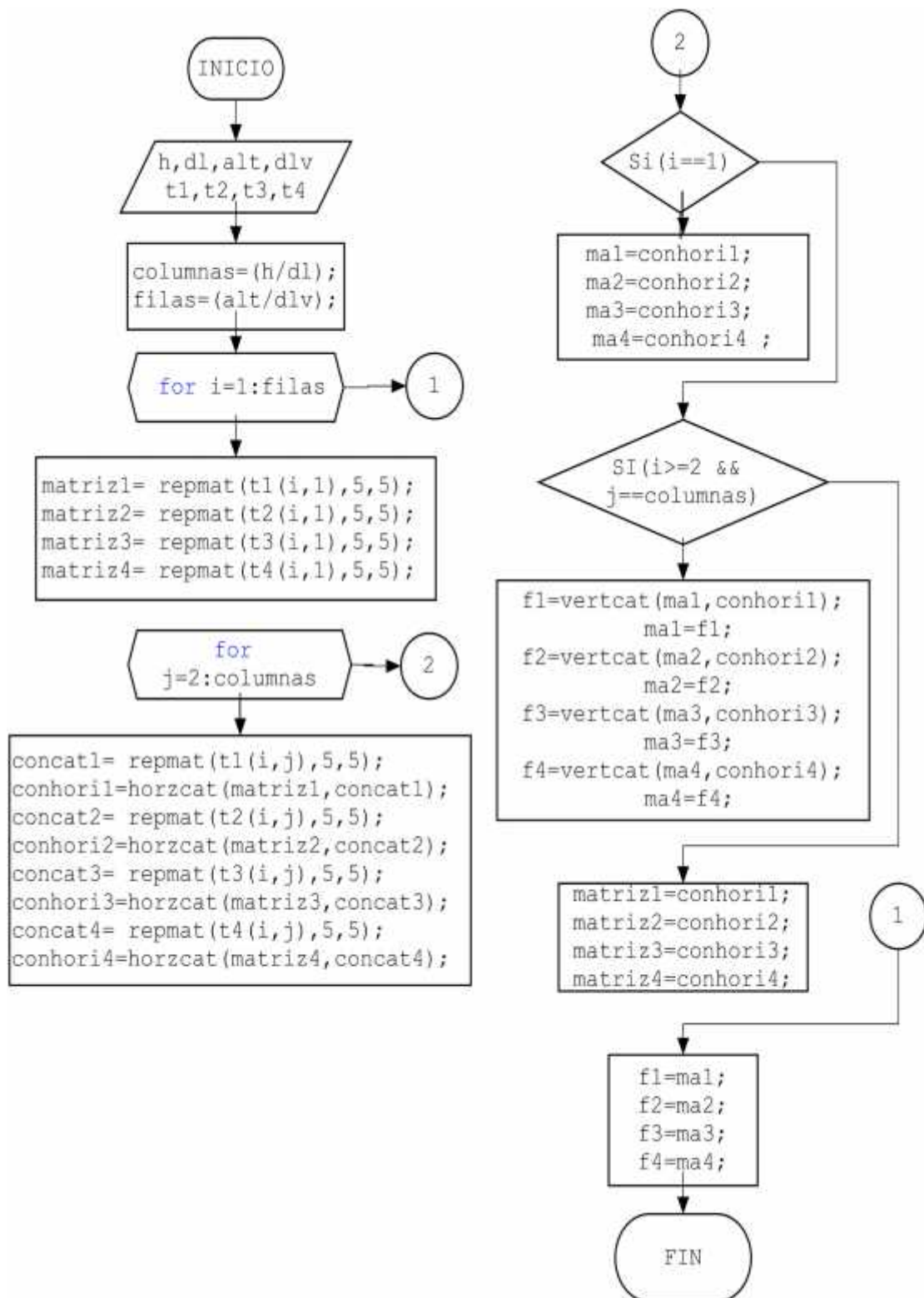
**Figura 23-3.** Reconstrucción 3D Objeto 2 con datos concatenados de **Tabla 7-3**.  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016



**Figura 24-3.** Reconstrucción 3D Objeto 2 con datos concatenados de **Tabla 9-3**.  
Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Finalmente en la **Figura 25-3** se expone el flujograma para realizar la concatenación de los datos y en el **Anexo C** se muestra el flujograma completo para el programa “Sistema Radar para Reconstrucción de Objetos 3D Mediante la Utilización del Dispositivo Kinect”.





**Figura 25-3.** Flujograma para concatenación de datos.  
**Fuente:** BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

### 3.6 Resolución y Precisión.

Para determinar la Resolución de los datos de profundidad que obtiene el Kinect, se tomó 11 mediciones experimentales en un objeto ubicado a 1000mm de distancia con respecto al Kinect, desplazando el mismo 10mm descendientemente hasta llegar a una distancia de 900mm a fin de obtener el Error Verdadero de cada dato y la media del Error Relativo Porcentual de los datos.

$$Ev = |\text{Valor Real} - \text{Valor Aproximado}| \quad \text{Ecuación. 1-3}$$

$$Et\% = \frac{\text{Error Verdadero}}{\text{Valor Verdadero}} \times 100\% \quad \text{Ecuación. 2-3}$$

$$\overline{Et} = \frac{Et_1 + Et_2 + Et_3 + \dots + Et_n}{n} \quad \text{Ecuación. 3-3}$$

Donde,  $Ev$  es el Error Verdadero,  $Et\%$  es el Error Relativo Porcentual y  $\overline{Et}$  es la media del Error Relativo Porcentual para  $n$  muestras (ver **Ecuación. 1-3**, **Ecuación. 2-3** y **Ecuación. 3-3**)

En la **Tabla 10-3** se muestra los datos de profundidad reales y obtenidos en las mediciones expresadas en milímetros además de mostrar El error verdadero para cada dato, el Error relativo porcentual y la media de los Errores Relativos.

**Tabla 10-3:** Errores de Profundidad.

Real	Medido	$Ev$	$Et$	$Et\%$
1000	1003	3	0,0030	0,300%
990	991	1	0,0010	0,101%
980	983	3	0,0031	0,306%
970	972	2	0,0021	0,206%
960	961	1	0,0010	0,104%
950	950	0	0,0000	0,000%
940	940	0	0,0000	0,000%
930	932	2	0,0022	0,215%
920	920	0	0,0000	0,000%
910	910	0	0,0000	0,000%
900	901	1	0,0011	0,111%
Media Error Relativo %				0,122%

Fuente: BARBERAN., Jeneffer & DOMÍNGUEZ, Rodrigo, 2016

Ahora bien la Resolución se define como a qué tan cercano está el valor medido del valor verdadero

De manera que para las mediciones que muestra la **Tabla 8-3** se determina que la resolución con la que se obtiene un dato de profundidad es de 3mm.

Seguidamente se calcula la media del error relativo porcentual para determinar la precisión de profundidad dando como resultado que el Kinect tiene un 99.88% de precisión para cuando la cámara IR y RGB se encuentren ubicadas en un Angulo paralelo al suelo.

$$\text{Precisión} = 100\% - \overline{Ert} \quad \text{Ecuación. 4-3}$$

$$\text{Precisión} = 100\% - 0.122\% \quad \text{Ecuación. 5-3}$$

$$\text{Precisión} = 99.88\% \quad \text{Ecuación. 6-3}$$

## CONCLUSIONES

El Sistema Radar para Reconstrucción de Objetos en 3D propuesto en este trabajo de investigación se desarrolla en cinco bloques principales que representan los procesos inherentes al manejo y uso de la información proporcionada por el dispositivo Kinect. Estos bloques son: Capturar la Imagen, Estimación de profundidad, Almacenar Información, Profundidad del Objeto y Reconstrucción del Objeto.

Se puede concluir que el motor de inclinación del sensor Kinect permite ajustar el ángulo de inclinación de la cámara IR y RGB en 27 grados hacia arriba y hacia abajo en 10 posiciones diferentes con un ángulo de calibración de  $\pm 5.4$  grados por cada posición, siendo 0 grados de inclinación cuando las cámaras mencionadas se encuentren alineadas paralelamente al eje del suelo. Esta tesis se desarrolló utilizando 0 grados de inclinación para obtener datos más precisos de medición.

Las pruebas realizadas para reconstruir objetos se realizaron dentro de un área de  $(2 \times 2)$  m<sup>2</sup> y se determinó que la profundidad de un objeto se obtiene a una distancia máxima de 1.2m y a una distancia mínima de 0.5m cuando la cámara de profundidad se encuentra paralela al eje del suelo.

El algoritmo diseñado, que fue implementado en Matlab, permite reconstruir objetos a través de matrices cuadradas binarias, se crea una matriz por cada fila existente en las matrices de profundidad, Así pues cada elemento de la matriz binaria con valor 1 representa una fracción del objeto.

La medida de profundidad que se obtiene sobre un objeto mediante la utilización del dispositivo Kinect tiene un margen de Error de 3mm y una precisión de 99.88%, valores obtenidos experimentalmente luego de las mediciones realizadas con objetos lineales.

La función `boxplot3` utilizada en el programa representa exaedros a cada elemento con valor 1 de la matriz binaria resultante, de manera que de acuerdo a las pruebas realizadas se concluye que debe existir un gran número de mediciones para que los exaedros sean imperceptibles al momento de visualizar el objeto reconstruido, Sin embargo, la concatenación de los datos en las matrices de profundidad propuesta en esta investigación permite que cada exaedro se asemeje a un punto en el diagrama de Reconstrucción 3D.



## RECOMENDACIONES

Se recomienda que las mediciones se deben realizar en desplazamientos verticales y horizontales, en intervalos cortos desde 3mm en adelante ya que a mayor número de mediciones se tendrá mejor precisión en la reconstrucción del objeto.

El Sistema Radar se puede hacer automático se realiza la implementación de un sistemas de control automático lo que permitiría mayor precisión en las mediciones de desplazamiento vertical y horizontal. Además en el momento de la adquisición de datos, al desplazar el Kinect horizontal y verticalmente mientras se capturan muestras, se deben evitar movimientos bruscos que cambien el grado de inclinación de la cámara IR, pues estos movimientos dificultan considerablemente la precisión en la toma de medidas.

Recomendamos el mejoramiento del algoritmo propuesto en esta investigación para Reconstruir Objetos en forma tridimensional, aumentando el número de vistas para la toma de mediciones con la finalidad de obtener mayor cantidad de datos de profundidad que permitan aumentar la precisión en la digitalización 3D de los objetos.

El algoritmo propuesto para la Reconstrucción de Objetos en 3D funciona correctamente para cualquier objeto sin importar el tipo de superficie que tenga, Sin embargo se recomienda que para la utilización del dispositivo Kinect en el Sistema Radar, no se debe utilizar objetos con superficies refractantes ni brillosos ya que estas superficies difuminan las ondas que propaga el Infrarrojo del Kinect de forma que se pierde información en las mediciones realizadas para ellos recomendamos utilizar objetos con superficies reflejantes para obtener datos exactos.

## BIBLIOGRAFIA

**ABHIJIT, Jana.** Kinect for Windows SDK Programming Guide. 1a ed. Birmingham – Mumbai. Packt Publishing Ltd. 2012. pp. 7-45, 77-155.

**ÁLVAREZ, René & SHAGÑAY, Sandra.** Sistema Radar Para Análisis Y Reconstrucción De Perfiles Topográficos A Media Escala. **TESIS.** Escuela Superior Politécnica de Chimborazo. Facultad de Informática y Electrónica. Escuela de Ingeniería Electrónica, Telecomunicaciones y Redes. Riobamba-Ecuador. 2015. p. 18.

### APRENDA MATLAB 4.2 COMO SI ESTUVIERA EN PRIMERO

<http://www4.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/Matlab42/matlab42.pdf>  
2015-11-20

### ARRAY CREATION AND CONCATENATION

<http://www.mathworks.com/help/matlab/elementary-matrices-and-arrays.html>  
2015-12-12

**BASSEM, Mahafza.** Analysis and Design using Matlab. 3a ed. Alabama-Estados Unidos. CRC Press. 2013. pp. 4-8.

**URL:** [http://dsp-book.narod.ru/RSAD/C1828\\_PDF\\_TOC.pdf](http://dsp-book.narod.ru/RSAD/C1828_PDF_TOC.pdf)  
Consultado 10-01-2016.

**BEYENIR, Sherat.** A Brief Introduction to Engineering Computation with MATLAB. 1a ed. OpenStax-CNX. 2014. pp. 5-19, 27-156.

**URL:** <https://cnx.org/a-brief-introduction-to-engineering-computation-with-matlab-9.2.pdf>  
Consultado 4-01-2016.

**BÖHLER, Wolfgang et al.** 3D scanning instruments. 1a ed. Thessaloniki-Grecia. Cipa. 2002. pp. 9-18.

**URL:** [https://i3mainz.hs-mainz.de/sites/default/files/public/data/p05\\_Boehler.pdf](https://i3mainz.hs-mainz.de/sites/default/files/public/data/p05_Boehler.pdf)  
Consultado 10-11-2015.

**CHUYA et al.** Diseño e Implementación de un Sistema para el Análisis del Movimiento Humano usando Sensores Kinect. **TESIS.** Universidad Politécnica Salesiana. Facultad de Ingeniería Electrónica. Cuenca-Ecuador. 2013. pp. 13-16.

**URL:** <http://dspace.ups.edu.ec/handle/123456789/5092>

Consultado 25-01-2016.

**CHAPRA, Steven & CANALE, Raymond.** Métodos Numéricos para ingenieros. 5a ed. Distrito Federal-México. McGraw-Hill. 2007. pp. 56-59.

**GARCIA, Adrian.** Estimación de zonas cruzables utilizando el sensor Kinect. **TESIS.** Universidad Carlos III de Madrid. Escuela Electrónica Industrial y Automática. 2012. pp. 12-27.

**URL:** [http://e-archivo.uc3m.es/bitstream/16876/Adrian\\_Garcia\\_Lopez.pdf?sequence=1](http://e-archivo.uc3m.es/bitstream/16876/Adrian_Garcia_Lopez.pdf?sequence=1).

Consultado 10-10-2015.

**GILAT, Amos.** Matlab: una Introducción con ejemplos prácticos. 2a ed. Barcelona-España. Reverte. 2005. pp. 5-247.

**URL:** [http://www.upibi.net/pdf//MATLAB\\_Una\\_introduccion\\_con\\_ejemplos\\_practicos.pdf](http://www.upibi.net/pdf//MATLAB_Una_introduccion_con_ejemplos_practicos.pdf)

Consultado 15-01-2016.

### **GRÁFICOS 3D EN MATLAB**

[http://personales.unican.es/corcuerp/Matlab\\_Simulink/Slides/Matlab\\_graficos3D.pdf](http://personales.unican.es/corcuerp/Matlab_Simulink/Slides/Matlab_graficos3D.pdf)

2015-11-05

**GUAMAN, Klever & SOSA, Andrés.** Diseño e Implementación de un Sistema para Digitalización en 3D mediante el uso de Matlab. **TESIS.** Universidad de las Fuerzas Armadas. Departamento de Eléctrica y Electrónica. Carrera de Ingeniería en Electrónica, Automatización y Control. Sangolquí-Ecuador. 2013. pp. 6-21.

**URL:** <http://repositorio.espe.edu.ec/xmlui/handle/21000/7629>

Consultado 17-01-2016.

**HERNANDEZ, Luis & HERRERA, Juan.** Análisis y Estudio de los Códigos Fuente SDK(Kit de Desarrollo de Software) e Implementación de una Aplicación Demostrativa que Registre la Captación de Movimiento de Manos y Brazos del Cuerpo Humano a través de Led's Indicadores mediante la utilización del Sensor Kinect del Xbox 360. **TESIS.** Universidad Politécnica Salesiana. Facultad de Ingeniería Electrónica. Escuela de Sistemas Industriales. Quito-Ecuador. 2013. pp. 9-43.

**URL:** <http://dspace.ups.edu.ec/handle/123456789/4143>

Consultado: 20-01-2016.

### **INTRODUCCIÓN A LOS ARCHIVOS MEX**

[http://www.mathworks.com/help/matlab/matlab\\_external/introducing-mex-files.html](http://www.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html)

2015-10-30

### **INTRODUCCIÓN A MATLAB**

<http://www.araba.ehu.es/depsi/jg/imatlab.pdf>

2015-11-05

### **KINECTFORMATLAB**

<http://kinectformatlab.es.tl/Home.htm>

2015-10-21

### **KINECT FOR MATLAB**

<http://www.mathworks.com/matlabcentral/linkexchange/links/2718-kinect-for-matlab>

2015-10-25

### **KINECT FOR WINDOWS SENSOR**

<https://msdn.microsoft.com/en-us/library/hh855355.aspx>

2015-10-10

### **KINECT MATLAB**

<http://www.mathworks.com/matlabcentral/fileexchange/30242-kinect-matlab>

2015-10-28

**MELVIN, William & SCHEER, James.** Principles of Modern Radar: Radar Applications. 1a ed. Stevenage-Reino Unido. Scitech Publishing. 2014. pp. 1-15.

**URL:** <http://www.theiet.org/resources/books/rsna/pmfv.cfm>

Consultado 02-02-2016.

**NUÑO, Simón.** Reconocimiento de Objetos mediante Sensor 3D Kinect. **TESIS.** Universidad Carlos III. Departamento de Ingeniería de Sistemas y Automática. Escuela de Ingeniería Electrónica y Automática. Leganés-Madrid. 2012. pp. 8-11,29-34.

**URL:** <http://docplayer.es/2389440-Reconocimiento-de-objetos-mediante-sensor-3d-kinect.html>

Consultado 10-10-2015.

#### **OBSERVATORIO LIDAR–CEFOP**

[http://www.tigo.cl/documents/sirgas/Poster\\_Lidar\\_Observatorio-LIDAR-CEFOP.pdf](http://www.tigo.cl/documents/sirgas/Poster_Lidar_Observatorio-LIDAR-CEFOP.pdf)

2015-11-20

#### **ONDAS Y RANGOS DE FRECUENCIA**

<http://www.radartutorial.eu/07.waves/pic/radarfrequencies.print.png>

2015-10-15

#### **OPENNI-WINDOWS-DRIVERS**

<https://github.com/benMcChesney/OpenNI-Windows-Drivers>

2015-10-02

#### **PRÁCTICAS DE AMPLIACIÓN DE MÉTODOS NUMÉRICOS CON MATLAB**

[http://www.ehu.eus/izaballa/Ana\\_Matr/Matlab/guia.pdf](http://www.ehu.eus/izaballa/Ana_Matr/Matlab/guia.pdf)

2015-11-15

**PEREZ, Evelin.** Reconocimiento de Movimientos del Cuerpo Humano basado en Tecnología Kinect. **TESIS.** Universidad Tecnológica Equinoccial. Facultad de Ciencias de la Ingeniería. Carrera de Ingeniería Mecatrónica. Quito-Ecuador. 2014. pp. 10-13, 20-25, 27, 44.

**URL:** <http://repositorio.ute.edu.ec/handle/123456789/5773>

Consultado 14-01-2015.

**QUINTANA, Matías.** Registro de una Secuencia Temporal de Nubes de Puntos utilizando Tecnología Kinect para la Reconstrucción Tridimensional de Material Arqueológico. **TESIS.** Pontificia Universidad Católica del Perú. Facultad de Ciencias e Ingeniería. Escuela de Ingeniería Electrónica. Lima-Perú. 2014. pp. 2-6, 11-15.

**URL:** <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6076>

Consultado 12-01-2015.

**SANCHEZ, Roberto.** Técnicas Algorítmicas para Cancelación de Ruido en Imágenes Radar. **TESIS.** Instituto Politécnico Nacional. Centro de Investigación en Computación. Maestría en Ciencias en Ingeniería de Computo Distrito Federal-México. 2007. pp. 4-6.

**URL:** <http://www.saber.cic.ipn.mx/cake/SABERsvn/trunk/Repositorios/webVerArchivo/409/2>

Consultado 12-02-2015.

**SEAN, Jonathan & HALL, Perry.** Met the Kinect: An Introduction to Programming Natural User Interfaces. 1a ed. New York-Estados Unidos. Apress. 2011. pp. 63-174

**URL:** <http://www.apress.com/9781430238881>

Consultado 05-01-2015.

#### **SISTEMA DE RADAR**

<http://webtelecomm.wordpress.com/2015/01/24/sistemas-de-radar/>

2015-12-03

**SKOLNIK, Merrill.** Introduction to Radar Systems. 3a ed. New York-Estados Unidos. McGraw Hill. 2011. Páginas 1-29.

#### **TIPOS DE RADAR**

<http://electromagnetismo2009.blogspot.com/2009/05/3.html>

2015-12-05

#### **VEC2MAT**

<http://www.mathworks.com/help/comm/ref/vec2mat.html>

2015-12-15

**YÁNEZ, Jorge.** Diseño y Construcción de un Escáner Bifocal para la Reconstrucción Visual de Objetos utilizando Procesamiento de Imágenes mediante Filtrado en Labview y Matlab.

**TESIS.** Universidad Politécnica Salesiana. Facultad de Ingeniería Electrónica. Quito-Ecuador. 2014. p. 16.

**URL:** <http://dspace.ups.edu.ec/handle/123456789/6861>

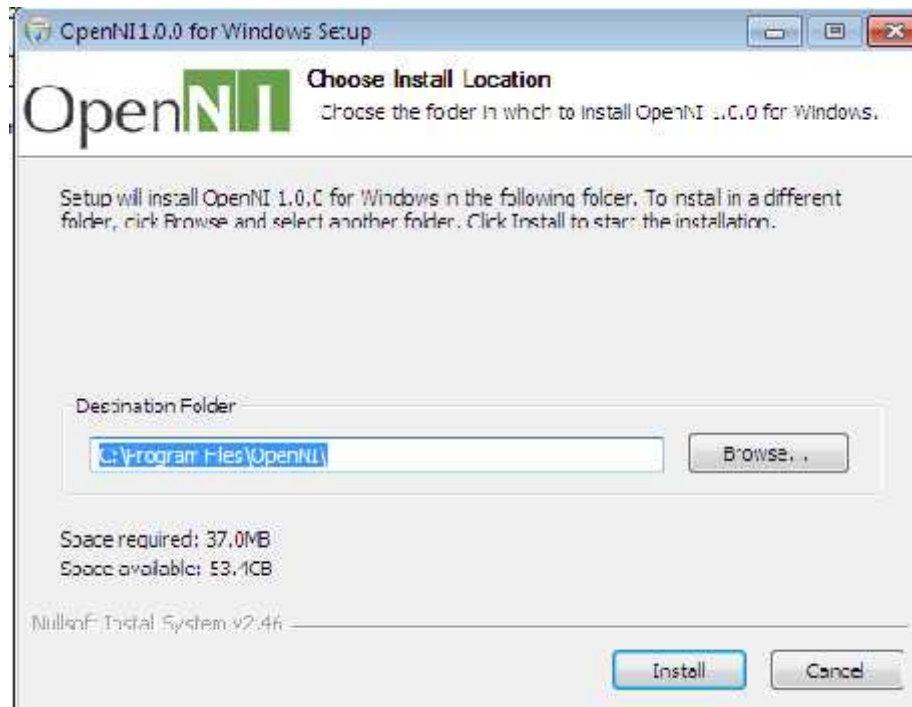
Consultado 10-11-2015.

## ANEXO A

### INSTALACION

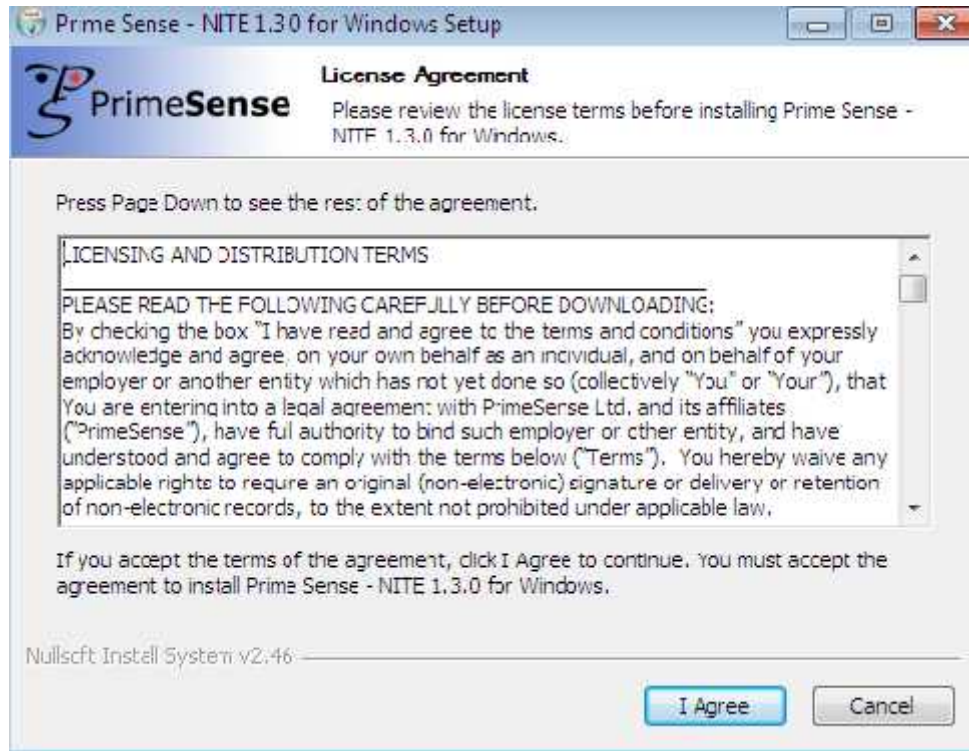
Para enlazar el dispositivo Kinect con matlab se requiere la instalación de los siguientes softwares.

#### ❖ Controlador Open NI.

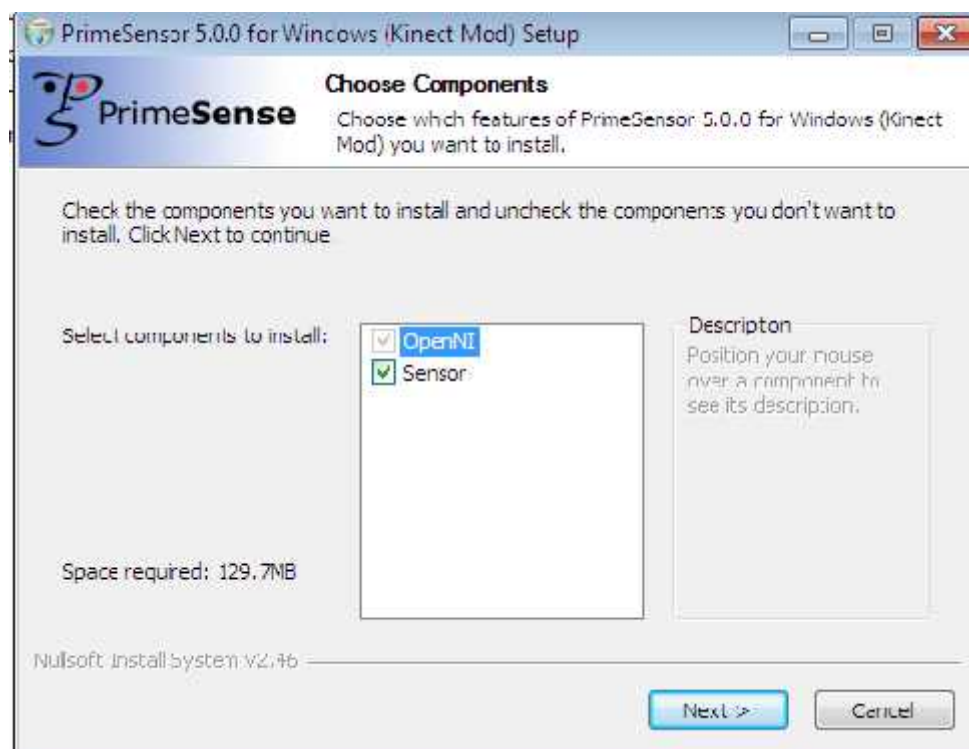




❖ **Librería NITE.**



❖ **Librería Sensor.**



## ANEXO B

### CODIGO SISTEMA RADAR COMPLETO

```
function varargout = CodigoCorregido2(varargin)

% INICIALIZACION DEL CODIGO

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @CodigoCorregido2_OpeningFcn, ...
                  'gui_OutputFcn',    @CodigoCorregido2_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% FIN DE LA INICIALIZACION DEL CODIGO


function CodigoCorregido2_OpeningFcn(hObject, eventdata, handles,
varargin)

% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to prueba6 (see VARARGIN)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CodigoCorregido2_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% % Se llama a la función altura que corresponde a la medida de la
arista

function altura_Callback(hObject, eventdata, handles)
```

```

% hObject      handle to altura (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of altura as text
% str2double(get(hObject,'String')) returns contents of altura as a
double

h = str2double(get(hObject,'String'));
if isnan(h) || ~isreal(h)
    set(handles.inicio,'String','Cannot plot f1')
    set(handles.inicio,'Enable','off')
    uicontrol(hObject)
else
    % Enable the Plot button with its original name
    set(handles.inicio,'String','Inicio')
    set(handles.inicio,'Enable','on')
end

% --- Executes during object creation, after setting all properties.
function altura_CreateFcn(hObject, eventdata, handles)
% hObject      handle to altura (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Se llama a la función delta que corresponde al desplazamiento
horizontal para las mediciones

function delta_Callback(hObject, eventdata, handles)

% hObject      handle to delta (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of delta as text
%         str2double(get(hObject,'String')) returns contents of delta
as a double

dl = str2double(get(hObject,'String'));
if isnan(dl) || ~isreal(dl)
    set(handles.inicio,'String','Cannot plot f1')
    set(handles.inicio,'Enable','off')
    uicontrol(hObject)
else
    set(handles.inicio,'String','Inicio')
    set(handles.inicio,'Enable','on')
end

```

```
% --- Executes during object creation, after setting all properties.
function delta_CreateFcn(hObject, eventdata, handles)
% hObject    handle to delta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

% Se llama a la función delta que corresponde al desplazamiento vertical para las mediciones

```
function deltav_Callback(hObject, eventdata, handles)
% hObject    handle to deltav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of deltav as text
%       str2double(get(hObject,'String')) returns contents of deltav
as a double
dlv = str2double(get(hObject,'String'));
```

```
if isnan(dlv) || ~isreal(dlv)
    set(handles.inicio,'String','Cannot plot f1')
    set(handles.inicio,'Enable','off')
    uicontrol(hObject)
else
    set(handles.inicio,'String','Inicio')
    set(handles.inicio,'Enable','on')
end
```

```
% --- Executes during object creation, after setting all properties.
function deltav_CreateFcn(hObject, eventdata, handles)
% hObject    handle to deltav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

% Se llama a la función alturaobjeto que corresponde a la altura del objeto

```
function alturaobjeto_Callback(hObject, eventdata, handles)
```

```

% hObject      handle to alturaobjeto (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of alturaobjeto as
text
%           str2double(get(hObject,'String')) returns contents of
alturaobjeto as a double
alt= str2double(get(hObject,'String'));

if isnan(alt) || ~isreal(alt)
    set(handles.inicio,'String','Cannot plot f1')
    set(handles.inicio,'Enable','off')
    uicontrol(hObject)
else
    % Enable the Plot button with its original name
    set(handles.inicio,'String','Inicio')
    set(handles.inicio,'Enable','on')
end

% --- Executes during object creation, after setting all properties.
function alturaobjeto_CreateFcn(hObject, eventdata, handles)
% hObject      handle to alturaobjeto (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in inicio.
function inicio_Callback(hObject, eventdata, handles)
% hObject      handle to inicio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)addpath(' ./Mex');
addpath(' ./Mex')
h = str2double(get(handles.altura,'String')); % Longitud cara lateral
cubo
dl = str2double(get(handles.delta,'String')); % distancia con se se
desplazara horizontal kinect para medicion
dlv = str2double(get(handles.deltav,'String')); % distancia con se se
desplazara horizontal kinect para medicion
alt = str2double(get(handles.alturaobjeto,'String')); %altura objeto

    choicel = questdlg('Registrar Primer Cara', 'Opciones',
'Registrar','Salir','Salir');
    switch choicel
        case 'Registrar'
            compl = 1;
        case 'Salir'
            close all;
    end
end

```

```

if comp1==1

    i=0;
    l=dl;
    base=0;
    alturaob=0;
    intervalo=h/dl;
    cont=1;
    intervaloalt=alt/dlv;
    contalt=1;

axis off;

    while (comp1==1 && base<=intervalo && alturaob<=intervaloalt)

        context = mxNiCreateContext('./Config/SamplesConfig.xml'); %%
        crear contexto con archivo xml Conecta el Microsoft Kinect Hardware
        width = 640; height = 480; %% inicializar figura
        fil = imagesc(zeros(height,width,'uint16'));% figura de
        profundidad
        for k=1:30
            option.adjust_view_point = true; %alinear
            mxNiUpdateContext(context, option);
            [rgb, depth] = mxNiImage(context); % Adquirir rgb
            y profundidad
            real_XYZ =
            mxNiConvertProjectiveToRealWorld(context, depth); % convertir
            uvZ[pix,pix,mm] a XYZ[mm,mm,mm]
            set(fil, 'CData', depth); % actualizar figura
            tmp = real_XYZ(:);% actualizar figura XYZ
            tmp(find(real_XYZ==0)) = NaN;
            tmp = reshape(tmp,[height width 3]);

            end
            handles.sencam=fil; %muestra figura de profundidad
            mxNiDeleteContext(context); %% borrar el objeto de contexto
            i=i+1;
            real=(real_XYZ(240,320,3)/1000); %seleccionar datos perfil if
            real<=h
                if (real<=h && real~=0 )
                    y(i)=h-real;
                else
                    y(i)=0;
                end
                x(i)=l;
                l=l+dl;
                cont=cont+1;
                base=cont;
                %-----Para Inicializar Fila siguiente-----
                if (base>intervalo && alturaob<intervaloalt)

                    choicel = questdlg('ir a la siguiente
                    fila','Opciones','Ir','Salir','Salir');
                    switch choicel
                        case 'Ir'
                            comp1 = 1;
                            cont=1;
                            l=dl;

```

```

                                base=0;
                                contalt=contalt+1;
                                alturaob=contalt;

                                case 'Salir'
                                    close all;
                                end
                            end
                        end
                    %-----

                    choice1 = questdlg('Registrar Siguiente
Dato', 'Opciones', 'Registrar', 'Salir', 'Salir');
                    switch choice1
                        case 'Registrar'
                            comp1 = 1;
                        case 'Salir'
                            close all;
                        end
                    end

                %Visualizar Tabla
                b=length(x);
                for j=1:b
                    tabla((j*2)-1)=x(j);
                end
                for j=1:b
                    tabla(2*j)=y(j);
                end
                a=0;
                for j=1:b
                    for s=1:2
                        a=a+1;
                        tablam(j,s)=tabla(a);
                    end
                end
                set(handles.uitable1, 'data', tablam)% visualizar tabla de valores

                %-----Transformacion Vector a Matriz
                columnas=(h/dl);
                t1=vec2mat(y,columnas) %Transformar de vector a matriz

            end

            %----CARA 2-----

            choice2 = questdlg('Registrar Segunda Cara', 'Opciones',
'Registrar', 'Salir', 'Salir');
            switch choice2
                case 'Registrar'
                    comp2 = 1;
                case 'Salir'
                    close all;
                end

            if comp2==1

                i=0;
                l=dl;

```

```

base=0;
alturaob=0;
intervalo=h/dl;
cont=1;
intervaloalt=alt/dlv;
contalt=1;

axis off;

while (comp2==1 && base<=intervalo && alturaob<=intervaloalt)

    context = mxNiCreateContext('./Config/SamplesConfig.xml'); %%
    crear contexto con archivo xml Conecta el Microsoft Kinect Hardware
    width = 640; height = 480; %% inicializar figura
    fil = imagesc(zeros(height,width,'uint16'));% figura de
    profundidad
        for k=1:30
            option.adjust_view_point = true; %alinear
            mxNiUpdateContext(context, option);
            [rgb, depth] = mxNiImage(context); % Adquirir rgb
            y profundidad
                real_XYZ =
                mxNiConvertProjectiveToRealWorld(context, depth); % convertir
                uvZ[pix,pix,mm] a XYZ[mm,mm,mm]
                set(fil, 'CData', depth); % actualizar figura
                tmp = real_XYZ(:);% actualizar figura XYZ
                tmp(find(real_XYZ==0)) = NaN;
                tmp = reshape(tmp,[height width 3]);
            end

            mxNiDeleteContext(context); %% borrar el objeto de contexto
            i=i+1;
            real=(real_XYZ(240,320,3)/1000); %seleccionar datos perfil if
            real<=h
                if (real<=h && real~=0 )
                    y(i)=h-real;
                else
                    y(i)=0;
                end
                x(i)=l;
                l=l+dl;
                cont=cont+1;
                base=cont;
                %-----Para Inicializar Fila siguiente-----
                if (base>intervalo && alturaob<intervaloalt)

                    choice2 = questdlg('ir a la siguiente
                    fila','Opciones','Ir','Salir','Salir');
                    switch choice2
                        case 'Ir'
                            comp2 = 1;
                            cont=1;
                            l=dl;
                            base=0;
                            contalt=contalt+1;
                            alturaob=contalt;
                        case 'Salir'
                            close all;

```



```

end
end
%-----

choice2 = questdlg('Registrar Siguiente
Dato','Opciones','Registrar','Salir','Salir');
switch choice2
case 'Registrar'
comp2 = 1;

case 'Salir'
close all;
end
end

%Visualizar Tabla
b=length(x);
for j=1:b
tabla((j*2)-1)=x(j);
end
for j=1:b
tabla(2*j)=y(j);
end
a=0;
for j=1:b
for s=1:2
a=a+1;
tablam(j,s)=tabla(a);
end
end
set(handles.uitable1, 'data', tablam)% visualizar tabla de valores

%-----Transformacion Vector a Matriz
columnas=(h/dl);
t2=vec2mat(y,columnas) %Transformar de vector a matriz

end
%-----FIN CARA 2-----

%-----INICIO CARA 3

choice3 = questdlg('Registrar Tercera Cara', 'Opciones',
'Registrar','Salir','Salir');
switch choice3
case 'Registrar'
comp3 = 1;
case 'Salir'
close all;
end

if comp3==1

i=0;
l=dl;
base=0;
alturaob=0;
intervalo=h/dl;
cont=1;

```

```

        intervaloalt=alt/dlv;
        contalt=1;

axis off;

        while (comp3==1 && base<=intervalo && alturaob<=intervaloalt)

                context = mxNiCreateContext('./Config/SamplesConfig.xml'); %%
crear contexto con archivo xml Conecta el Microsoft Kinect Hardware
                width = 640; height = 480; %% inicializar figura
                fil = imagesc(zeros(height,width,'uint16'));% figura de
profundidad
                for k=1:30
                        option.adjust_view_point = true; %alinear
profundidad en rgb
                                mxNiUpdateContext(context, option);
                                [rgb, depth] = mxNiImage(context); % Adquirir rgb
y profundidad
                                real_XYZ =
mxNiConvertProjectiveToRealWorld(context, depth); % convertir
uvZ[pix,pix,mm] a XYZ[mm,mm,mm]
                                set(fil, 'CData', depth); % actualizar figura
                                tmp = real_XYZ(:);% actualizar figura XYZ
                                tmp(find(real_XYZ==0)) = NaN;
                                tmp = reshape(tmp,[height width 3]);

                                end

                                mxNiDeleteContext(context); %% borrar el objeto de contexto
                                i=i+1;
                                real=(real_XYZ(240,320,3)/1000); %seleccionar datos perfil if
real<=h
                                        if (real<=h && real~=0 )
                                                y(i)=h-real;
                                        else
                                                y(i)=0;
                                        end
                                x(i)=1;
                                l=l+dl;
                                cont=cont+1;
                                base=cont;
                                %-----Para Inicializar Fila siguiente-----
                                        if (base>intervalo && alturaob<intervaloalt)

                                                choice3 = questdlg('ir a la siguiente
fila','Opciones','Ir','Salir','Salir');
                                                switch choice3
                                                        case 'Ir'
                                                                comp3 = 1;
                                                                cont=1;
                                                                l=dl;
                                                                base=0;
                                                                contalt=contalt+1;
                                                                alturaob=contalt;

                                                        case 'Salir'
                                                                close all;
                                                        end
                                                end
                                end
                                %-----

```

```

        choice3 = questdlg('Registrar Siguiente
Dato','Opciones','Registrar','Salir','Salir');
        switch choice3
            case 'Registrar'
                comp3 = 1;
            case 'Salir'
                close all;
            end
        end

end

%Visualizar Tabla
b=length(x);
for j=1:b
    tabla((j*2)-1)=x(j);
end
for j=1:b
    tabla(2*j)=y(j);
end
a=0;
for j=1:b
    for s=1:2
        a=a+1;
        tablam(j,s)=tabla(a);
    end
end
set(handles.uitable1, 'data', tablam)% visualizar tabla de valores

%-----Transformacion Vector a Matriz
columnas=(h/dl);
t3=vec2mat(y,columnas) %Transformar de vector a matriz

end

%-----FIN CARA 3-----

%----CARA 4-----

        choice4 = questdlg('Registrar Cuarta Cara', 'Opciones',
'Registrar','Salir','Salir');
        switch choice4
            case 'Registrar'
                comp4 = 1;
            case 'Salir'
                close all;
            end
        end

if comp4==1

    i=0;
    l=dl;
    base=0;
    alturaob=0;
    intervalo=h/dl;
    cont=1;
    intervaloalt=alt/dlv;
    contaalt=1;

axis off;

```

```

while (comp4==1 && base<=intervalo && alturaob<=intervaloalt)

    context = mxNiCreateContext('./Config/SamplesConfig.xml'); %%
    crear contexto con archivo xml Conecta el Microsoft Kinect Hardware
    width = 640; height = 480; %% inicializar figura
    fil = imagesc(zeros(height,width,'uint16'));% figura de
profundidad
    for k=1:30
        option.adjust_view_point = true; %alinear
profundidad en rgb
        mxNiUpdateContext(context, option);
        [rgb, depth] = mxNiImage(context); % Adquirir rgb
y profundidad
        real_XYZ =
mxNiConvertProjectiveToRealWorld(context, depth); % convertir
uvZ[pix,pix,mm] a XYZ[mm,mm,mm]
        set(fil, 'CData', depth); % actualizar figura
        tmp = real_XYZ(:);% actualizar figura XYZ
        tmp(find(real_XYZ==0)) = NaN;
        tmp = reshape(tmp,[height width 3]);

    end

    mxNiDeleteContext(context); %% borrar el objeto de contexto
    i=i+1;
    real=(real_XYZ(240,320,3)/1000); %seleccionar datos perfil if
real<=h
        if (real<=h && real~=0 )
            y(i)=h-real;
        else
            y(i)=0;
        end
        x(i)=l;
        l=l+dl;
        cont=cont+1;
        base=cont;
%-----Para Inicializar Fila siguiente-----
        if (base>intervalo && alturaob<intervaloalt)

            choice4 = questdlg('ir a la siguiente
fila','Opciones','Ir','Salir','Salir');
            switch choice4
                case 'Ir'
                    comp4 = 1;
                    cont=1;
                    l=dl;
                    base=0;
                    contalt=contalt+1;
                    alturaob=contalt;
                case 'Salir'
                    close all;
            end
        end
%-----

        choice4 = questdlg('Registrar Siguiente
Dato','Opciones','Registrar','Salir','Salir');
        switch choice4
            case 'Registrar'

```

```

        comp4 = 1;

        case 'Salir'
            close all;
        end
    end

end

%Visualizar Tabla
    b=length(x);
    for j=1:b
        tabla((j*2)-1)=x(j);
    end
    for j=1:b
        tabla(2*j)=y(j);
    end
    a=0;
    for j=1:b
        for s=1:2
            a=a+1;
            tablam(j,s)=tabla(a);
        end
    end
set(handles.uitable1, 'data', tablam)% visualizar tabla de valores

%-----Transformacion Vector a Matriz
columnas=(h/dl);
t4=vec2mat(y,columnas) %Transformar de vector a matriz
end

%-----FIN CARA 4-----

%----- Concatenacion Datos 5*5 Matrices caras laterales

columnas=(h/dl);
filas=(alt/dlv);

    for i=1:filas %filas
        matriz1= repmat(t1(i,1),5,5); % repmat replica dato de
matriz1 a (n*n)
        matriz2= repmat(t2(i,1),5,5); % repmat replica dato de
matriz2 a (n*n)
        matriz3= repmat(t3(i,1),5,5); % repmat replica dato de
matriz3 a (n*n)
        matriz4= repmat(t4(i,1),5,5); % repmat replica dato de
matriz4 a (n*n)
        for j=2:columnas %columnas
            concat1= repmat(t1(i,j),5,5);
            conhoril=horzcat(matriz1,concat1);% horzcat
concatenacion horizontal
            concat2= repmat(t2(i,j),5,5);
            conhoril2=horzcat(matriz2,concat2);%
            concat3= repmat(t3(i,j),5,5);
            conhoril3=horzcat(matriz3,concat3);%
            concat4= repmat(t4(i,j),5,5);
            conhoril4=horzcat(matriz4,concat4);%
            if(i==1)
                ma1=conhoril ;
                ma2=conhoril2 ;
                ma3=conhoril3 ;
            end
        end
    end
end

```

```

        ma4=conhori4 ;
    end
    if(i>=2 && j==columnas)
        f1=vertcat(ma1,conhori1);    % vertcat
concatenacion horizontal
        ma1=f1;
        f2=vertcat(ma2,conhori2);    %
        ma2=f2;
        f3=vertcat(ma3,conhori3);    %
        ma3=f3;
        f4=vertcat(ma4,conhori4);    %
        ma4=f4;

    end
    matriz1=conhori1;
    matriz2=conhori2;
    matriz3=conhori3;
    matriz4=conhori4;

    end
end
    f1=ma1 % Matriz Concatenada primera cara.
    f2=ma2 % Matriz Concatenada segunda cara.
    f3=ma3 % Matriz Concatenada tercera cara.
    f4=ma4 % Matriz Concatenada cuarta cara.

%-----GRAFICA BOXPLOT3-----

cuad=(columnas*5);
contfilas=1;
posz=0;
while contfilas<=intervaloalt*5
matcerosc1=zeros(cuad);
matcerosc2=zeros(cuad);
matcerosc3=zeros(cuad);
matcerosc4=zeros(cuad);

    for i=1:columnas*5
        posc1= f1(contfilas,i);
        posc11=((posc1*cuad)/(h*5));
        posc111=round(posc11)*5;

        posc2= f2(contfilas,i);
        posc22=((posc2*cuad)/(h*5));
        posc222=round(posc22)*5;

        posc3=f3(contfilas,i);
        posc33=((posc3*cuad)/(h*5));
        posc333=round(posc33)*5;

        posc4= f4(contfilas,i);
        posc44=((posc4*cuad)/(h*5));
        posc444=round(posc44)*5;

        for j=1:posc111
            matcerosc1(j,i)=1;
        end
    end
end

```

```

        for j=1:posc222
            matcerosc2(j,i)=1;
        end

        for j=1:posc333
            matcerosc3(j,i)=1;
        end

        for j=1:posc444
            matcerosc4(j,i)=1;
        end

    end
    cara4=rot90(matcerosc4,1);
    cara3=matcerosc3;
    cara2=rot90(matcerosc2,3);
    cara1=rot90(matcerosc1,2);

    for i=1:cuad
        for j=1:cuad
            A(i,j)=(cara1(i,j)*cara2(i,j)*cara3(i,j)*cara4(i,j));
        end
    end
    A = fliplr(A);

    dl1=0;
    dl2=0;
    for d=1:cuad
        for e=1:cuad
            if B(d,e)==1

                boxplot3C(dl1,dl2,posz,dl,dl,dlv,1)
                hold on;
                grid on;
                rotate3d;
            end
            dl1=dl1+dl;

        end
        dl2=dl2+dl;
        dl1=0;
    end
    contfilas=contfilas+1;
    posz=posz+dlv;
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all;

% --- Executes during object creation, after setting all properties.

```

```

function uitable1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
% callend

% --- Executes on button press in box.

% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in box.
function boxplot3(x0,y0,z0,Lx,Ly,Lz,w)
% hObject    handle to box (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

x= [x0, x0, x0, x0, (x0+Lx), (x0+Lx), (x0+Lx), (x0+Lx)]; %(1x8)
y= [y0, y0, y0+Ly, y0+Ly, y0, y0, y0+Ly, y0+Ly]; %(1x8)
z= [z0, z0+Lz, z0+Lz, z0, z0, z0+Lz, z0+Lz, z0]; %(1x8)
index = zeros(6,5);
index(1,:) = [1 2 3 4 1];
index(2,:) = [5 6 7 8 5];
index(2,:) = [5 6 7 8 5];
index(3,:) = [1 2 6 5 1];
index(4,:) = [4 3 7 8 4];
index(5,:) = [2 6 7 3 2];
index(6,:) = [1 5 8 4 1];
c='rgbcmy';
for k = 1:6

    if w~=0
        fill3(x(index(k,:)), y(index(k,:)), z(index(k,:)), c(k))
    else
        plot3(x(index(k,:)), y(index(k,:)), z(index(k,:)))
    end
end
end

```



## ANEXO C

### FLUJOGRAMA

